



Informatica® Multidomain MDM
10.3

Guía de servicios de entidad de negocio

Este software y la documentación se proporcionan exclusivamente en virtud de un acuerdo de licencia independiente que contiene restricciones de uso y divulgación. Ninguna parte de este documento puede ser reproducida o transmitida de cualquier forma o manera (electrónica, fotocopia, grabación o mediante otros métodos) sin el consentimiento previo de Informatica LLC.

Las bases de datos, el software y los programas de DERECHOS DEL GOBIERNO DE LOS ESTADOS UNIDOS, y la documentación e información técnica relacionadas entregadas a los clientes del Gobierno de los Estados Unidos constituyen "software informático comercial" o "datos técnicos comerciales" de acuerdo con el Reglamento de Adquisición Federal y las regulaciones complementarias específicas del organismo que correspondan. Como tales, el uso, la duplicación, la divulgación, la modificación y la adaptación están sujetos a las restricciones y los términos de licencia establecidos en el contrato gubernamental aplicable, y hasta donde sea aplicable en función de los términos del contrato gubernamental, a los derechos adicionales establecidos en FAR 52.227-19, Licencia de Software Informático Comercial.

Informatica, el logotipo de Informatica y ActiveVOS son marcas comerciales o marcas comerciales registradas de Informatica LLC en Estados Unidos y en las diversas jurisdicciones de todo el mundo. La lista actual de marcas comerciales de Informatica está disponible en Internet en <https://www.informatica.com/trademarks.html>. Otros nombres de productos y empresas pueden ser nombres o marcas comerciales de sus respectivos titulares.

Las partes de este software o la documentación están sujetas a derechos de autor de terceros. Se incluyen con el producto los avisos obligatorios de terceros.

La información contenida en esta documentación está sujeta a cambios sin previo aviso. Si encuentra algún problema en esta documentación, escríbanos a infa_documentation@informatica.com para notificarnoslo.

Los productos de Informatica gozan de garantía en función de los términos y condiciones de los acuerdos conforme a los cuales se proporcionen. **INFORMATICA PROPORCIONA LA INFORMACIÓN DE ESTE DOCUMENTO "TAL CUAL" SIN GARANTÍA DE NINGÚN TIPO, EXPRESA O IMPLÍCITA, INCLUIDAS LAS GARANTÍAS DE COMERCIALIZACIÓN, ADAPTACIÓN A UN FIN PARTICULAR Y CUALQUIER GARANTÍA O CONDICIÓN DE NO INCUMPLIMIENTO.**

Tabla de contenido

Prefacio	7
Documentación de Informática	7
Informatica Network.	7
Base de conocimiento de Informática.	7
Documentación de Informática.	7
Matrices de disponibilidad de productos de Informática.	8
Informatica Velocity.	8
Catálogo de soluciones de Informática.	8
Servicio internacional de atención al cliente de Informática.	8
Capítulo 1: Introducción a los servicios de entidad de negocio.....	9
Resumen de los servicios de entidad de negocio.	9
Servicios de entidad de negocio.	10
Servicio de entidad de negocio ReadBE.	10
Servicio de entidad de negocio WriteBE.	10
Servicio de entidad de negocio SearchBE.	11
Extremos del servicio de entidad de negocio.	11
Extremo Enterprise JavaBeans para servicios de entidad de negocio.	11
Extremo REST para servicios de entidad de negocio.	11
Llamadas del servicio de entidad de negocio de REST y EJB.	11
Extremo SOAP para servicios de entidad de negocio.	12
Identificar un registro raíz.	12
Filtros de datos y seguridad.	13
Capítulo 2: Llamadas de servicio de entidad de negocio Enterprise Java Bean.....	14
Resumen de las llamadas de servicio de entidad de negocio Enterprise Java Bean.	14
Ejemplo de código Java con clases SDO estándares.	14
Ejemplo de código Java con clases SDO generadas.	18
Capítulo 3: Llamadas del servicio de entidad de negocio de transferencia de estado representacional.....	23
Resumen de las API de REST para servicios de entidad de negocio.	23
Métodos REST compatibles.	24
Método de autenticación.	24
Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros.	24
Archivos de idioma de descripción de aplicación web.	25
Localizador uniforme de recursos de REST.	26
Configuración de encabezado y cuerpo.	27
Encabezado de solicitud.	27

Cuerpo de la solicitud.	28
Parámetros de consulta estándar.	29
Formatos de fecha y hora UTC.	30
Configuración de WebLogic para ejecutar llamadas REST de servicios de entidad de negocio.	31
Visualización de parámetros de entrada y salida.	31
Plantilla de JavaScript.	32
Ejemplo de JavaScript.	33
Referencia de API de REST para servicios de entidad de negocio	35
Obtener metadatos	35
Lista de metadatos.	38
Leer registros.	43
Crear registros.	50
Actualizar registros.	52
Eliminar registros.	55
Lista Registro.	56
Búsqueda Registro.	58
Proveedor de sugerencias.	64
Obtener metadatos de BPM.	65
Lista de tareas.	66
Leer tarea.	71
Crear tarea.	73
Actualizar tarea.	76
Tarea completada.	79
Ejecutar acción de tarea.	81
Lista de usuarios asignables.	83
Enumerar metadatos de archivo.	84
Crear metadatos de archivo.	85
Obtener metadatos de archivo.	86
Actualizar metadatos de archivo.	87
Cargar contenido de archivo.	88
Obtener contenido de archivo.	89
Suprimir archivo.	89
Vista previa de promoción.	90
Promover.	92
Eliminar pendiente.	92
Vista previa de fusión.	93
Fusión pendiente.	96
Promover fusión.	97
Fusionar registros.	98
Anular fusión de registros.	100
Leer una relación.	101
Crear una relación.	104

Actualizar una relación.	105
Eliminar una relación.	107
Obtener registros relacionados.	107
Leer registros coincidentes.	111
Actualizar registros coincidentes.	113
Eliminar registros coincidentes.	113
Obtener eventos de historial de registros.	114
Obtener detalles de evento.	117
Obtener metadatos de DaaS.	119
Búsqueda de DaaS.	120
Lectura de DaaS.	125
WriteMerge.	126
Importación de DaaS.	127
Actualización de DaaS.	130

Capítulo 4: Llamadas de servicios de entidad de negocio de Protocolo simple de acceso a objetos. 133

Llamadas de Protocolo simple de acceso a objetos para servicios de entidad de negocio.	133
Método de autenticación.	134
Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros.	134
Archivo de lenguaje de descripción de servicios web.	135
URL de SOAP.	136
Respuestas y solicitudes SOAP.	137
Visualización de parámetros de entrada y salida.	138
Referencia de API de SOAP.	139
Respuesta y solicitud SOAP de ejemplo.	140

Capítulo 5: Servicios para registros de referencias cruzadas y cálculos de BVT. 142

Resumen de los servicios para registros de referencias cruzadas y cálculos de BVT.	142
Obtención de datos de referencias cruzadas e investigación de cálculos de BVT.	142
Obtener registros de referencias cruzadas.	143
Determinar colaboradores del registro principal.	143
Obtener las puntuaciones de confianza de campos de registro de referencias cruzadas colaboradores.	144
Obtención de las puntuaciones de confianza de todos los campos de registro de referencias cruzadas.	145
Obtener información sobre los sistemas de origen.	146
Ejemplo de obtención de información sobre los sistemas de origen.	146
Filtrado y paginación de respuestas.	147
Ejemplos de solicitudes de filtrado.	147
Establecer la mejor versión de confianza.	147
Seleccionar el campo colaborador correcto.	147
Ejemplo de selección del campo colaborador correcto.	148

Escribir el valor correcto en el registro principal.	148
Ejemplo de escritura del valor correcto en el registro principal.	150
Eliminar los datos de origen no coincidentes.	151
Ejemplo de eliminación de datos de origen no coincidentes.	152
Respuesta de anulación de fusión.	153
Capítulo 6: Compatibilidad del servicio de vinculación empresarial.....	154
Resumen.	154
Servicios de entidad de negocio para la importación y actualización de DaaS.	154
Configuración de la compatibilidad de la vinculación.	155
Aplicación personalizada para la división de datos de vinculación.	155
Capítulo 7: Llamadas externas para limpiar, analizar y transformar datos... 156	
Resumen.	156
Eventos compatibles.	157
Cómo configurar llamadas externas.	157
Ejemplo: Validación y lógica personalizadas para servicios de entidad de negocio.	158
Requisitos previos.	158
Paso 1. Probar la validación personalizada.	158
Paso 2. Probar la lógica personalizada.	159
Apéndice A: Uso de las API de REST para añadir registros..... 164	
Resumen sobre el uso de las API de REST para añadir registros.	164
Estructura de una entidad de negocio de persona.	165
Paso 1. Obtener información sobre el esquema.	165
Obtener la respuesta de los metadatos.	166
Paso 2. Crear un registro.	171
Crear respuestas de registro.	172
Paso 3. Leer el registro.	173
Leer la respuesta del registro.	174
Apéndice B: Uso de las API de REST para cargar archivos..... 178	
Resumen sobre el uso de las API de REST para cargar archivos.	178
API de REST para archivos.	179
Componentes de archivo.	179
Tipos de almacenamiento.	180
Adjuntar archivos a registros.	181
Adjuntar archivos a tareas.	182
Cargar archivos de paquete de recursos.	185
Índice..... 186	

Prefacio

Bienvenido a la *Guía de servicios de entidad de negocio de Multidomain MDM*. En esta guía se explica cómo realizar llamadas de servicio de entidad de negocio para operar en entidades de negocio en Informatica® MDM Hub.

Esta guía está dirigida a los especialistas técnicos que se encargan de configurar interfaces de usuario personalizadas para realizar llamadas de entidad de negocio a MDM Hub.

Documentación de Informatica

Informatica Network

Informatica Network incluye el servicio internacional de atención al cliente de Informatica, la base de conocimiento de Informatica y otros recursos de producto. Para acceder a Informatica Network, visite <https://network.informatica.com>.

Un miembro puede:

- Acceder a todos sus recursos de Informatica en un solo lugar.
- Busque recursos de producto, como documentación, preguntas frecuentes y mejores prácticas en la base de conocimiento.
- Vea la información de disponibilidad del producto.
- Revisar los casos de asistencia.
- Buscar su red de grupos de usuarios de Informatica locales y colaborar con sus iguales.

Base de conocimiento de Informatica

Utilice la base de conocimiento de Informatica para buscar recursos de producto como documentación, artículos de procedimientos, mejores prácticas y PAM en la red de Informatica.

Para acceder a la base de conocimiento, visite <https://kb.informatica.com>. Si tiene preguntas, comentarios o ideas relacionadas con la base de conocimiento de Informatica, póngase en contacto con el equipo de la base de conocimiento de Informatica en KB_Feedback@informatica.com.

Documentación de Informatica

Para obtener la documentación más reciente del producto, consulte la base de conocimiento de Informatica en https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

Si tiene preguntas, comentarios o ideas relacionadas con esta documentación, póngase en contacto con el equipo de documentación de Informatica enviando un correo electrónico a infa_documentation@informatica.com.

Matrices de disponibilidad de productos de Informatica

Las matrices de disponibilidad de producto (PAM, Product Availability Matrixes) indican las versiones de sistemas operativos, bases de datos y otros tipos de orígenes de datos y destinos admitidos por una versión de un producto. Si es miembro de la red de Informatica, puede acceder a las PAM en <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity es un conjunto de sugerencias y mejores prácticas desarrollado por los servicios profesionales de Informatica. Desarrollado a partir de la experiencia real de cientos de proyectos de administración de datos, Informatica Velocity representa el conocimiento conjunto de nuestros asesores, los cuales han trabajado con organizaciones de todo el mundo para planificar, desarrollar, implementar y mantener con éxito soluciones de administración de datos.

Si es miembro de la red de Informatica, puede acceder a los recursos de Informatica Velocity en <http://velocity.informatica.com>.

Si tiene alguna pregunta, comentario o idea acerca de Informatica Velocity, póngase en contacto con los servicios Profesionales de Informatica en ips@informatica.com.

Catálogo de soluciones de Informatica

El Catálogo de soluciones de Informatica es un foro donde puede buscar soluciones que aumenten, amplíen o mejoren sus implementaciones de Informatica. Al aprovechar cualquiera de los cientos de soluciones de los desarrolladores y los socios de Informatica, puede mejorar la productividad y acelerar el tiempo de implementación en los proyectos. Puede acceder al Catálogo de soluciones de Informatica en <https://marketplace.informatica.com>.

Servicio internacional de atención al cliente de Informatica

Puede ponerse en contacto con un centro de atención global por teléfono o a través del soporte en línea en la red de Informatica.

Para encontrar el número de teléfono local del servicio internacional de atención al cliente de Informatica, visite el sitio web de Informatica en el siguiente vínculo:

<http://www.informatica.com/us/services-and-training/support-services/global-support-centers>.

Si es miembro de la red de Informatica, puede utilizar el soporte en línea en <http://network.informatica.com>.

CAPÍTULO 1

Introducción a los servicios de entidad de negocio

Este capítulo incluye los siguientes temas:

- [Resumen de los servicios de entidad de negocio, 9](#)
- [Servicios de entidad de negocio, 10](#)
- [Extremos del servicio de entidad de negocio, 11](#)
- [Identificar un registro raíz, 12](#)
- [Filtros de datos y seguridad , 13](#)

Resumen de los servicios de entidad de negocio

Un servicio de entidad de negocio es un conjunto de operaciones que ejecutan código de MDM Hub para crear, actualizar, eliminar y buscar registros de objeto base en una entidad de negocio. Puede desarrollar una interfaz de usuario personalizada que pueda ejecutar código Java o JavaScript para realizar llamadas de servicios de entidad de negocio.

Por ejemplo, puede crear un servicio de entidad de negocio para aumentar un registro de proveedor con datos de Dun and Bradstreet. Configure el servicio de entidad de negocio para tomar el registro de proveedor como una entrada, recuperar alguna información de Dun and Bradstreet, actualizar el registro y, después, generar el registro de proveedor actualizado.

Los objetos base de una entidad de negocio tienen los siguientes servicios de entidad de negocio:

Lectura

Cada entidad de negocio tiene un servicio de entidad de negocio para realizar una operación de lectura.

Escritura

Cada entidad de negocio tiene un servicio de entidad de negocio para realizar una operación de escritura.

Búsqueda

Cualquier entidad de negocio que tenga campos que admiten búsquedas tiene un servicio de entidad de negocio para realizar una operación de búsqueda.

Por ejemplo, una entidad de negocio de persona tiene campos que admiten búsquedas. MDM Hub genera un servicio de entidad de negocio ReadPerson, WritePerson y SearchPerson. Los pasos de los servicios de entidad de negocio de lectura, escritura y búsqueda le permiten leer, crear, actualizar, eliminar y buscar registros en un objeto compuesto.

Servicios de entidad de negocio

Un servicio de entidad de negocio realiza una operación. Puede usar los servicios de entidad de negocio ReadBE, WriteBE y SearchBE.

Un servicio de entidad de negocio cuenta con pasos de servicio. Una solicitud entrante pasa a través de cada paso de servicio. La salida de un paso es una entrada del paso siguiente. La salida de un paso puede pasar información a la entrada del siguiente paso. Todos los pasos de servicio de entidad de negocio se ejecutan como una llamada Enterprise Java Bean en una sola transacción. MDM Hub gestiona las excepciones.

Nota: Antes de usar los servicios de entidad de negocio, debe validar el Almacén de referencias operativas.

Servicio de entidad de negocio ReadBE

El servicio de entidad de negocio ReadBE lee los datos de un registro de objeto base en una entidad de negocio.

Puede especificar los parámetros de paginación con el paso ReadBE para establecer el número de registros que se van a devolver y la página de resultados que se van a ver.

Los resultados del servicio ReadBE no incluyen los registros que se hayan eliminado temporalmente.

Si no pasa el parámetro EffectiveDate en la solicitud de servicio de entidad de negocio, MDM Hub da por supuesto un valor NULL para la fecha de vigencia y el servicio de entidad de negocio lee los datos del objeto base. Si pasa el parámetro EffectiveDate, MDM Hub calculará la mejor versión de la verdad a partir de registros de referencias cruzadas y el servicio de entidad de negocio de lectura devuelve la mejor versión actualizada de la verdad.

Servicio de entidad de negocio WriteBE

El servicio de entidad de negocio WriteBE puede actualizar los datos de un elemento de entidad de negocio, crear un elemento secundario de entidad de negocio o eliminar elementos secundarios de entidad de negocio.

Nota: El servicio de entidad de negocio WriteBE utiliza la configuración de confianza existente para calcular la confianza en los objetos base. Con este servicio no puede realizar un reemplazo de confianza.

Parámetros opcionales

En la siguiente tabla se describen los parámetros opcionales que puede usar con el servicio de entidad de negocio WriteBE:

Parámetro	Descripción
recordState	Establezca el estado del registro en ACTIVE, PENDING o DELETED. Nota: Cuando se establece recordState=ACTIVE y se ejecuta el servicio en un registro eliminado temporalmente, el servicio devuelve el registro al estado activo.
EffectivePeriod	Especifique un período efectivo. Si no pasa el parámetro EffectivePeriod, MDM Hub asumirá que es un periodo sin enlazar. MDM Hub no comprueba que exista una alineación para los periodos efectivos entre los objetos raíz y los objetos secundarios. Al crear o actualizar los registros, asegúrese de los periodos efectivos del registro principal y los registros secundarios estén alineados.

Servicio de entidad de negocio SearchBE

Utilice el servicio de entidad de negocio SearchBE para buscar un registro raíz en una entidad de negocio.

Para obtener información sobre la configuración de entidades de negocio para Smart Search, consulte la *Guía de configuración de Multidomain MDM*.

Extremos del servicio de entidad de negocio

Es posible acceder a los servicios de entidad de negocio a través de un extremo Enterprise JavaBeans (EJB), un extremo de transferencia de estado representacional (REST) o un extremo de Protocolo simple de acceso a objetos (SOAP).

Los extremos REST se basan en una perspectiva de EJB. La configuración de los servicios de entidad de negocio de REST define cómo se asignan las URL de REST a las llamadas de servicio de entidad de negocio EJB.

Extremo Enterprise JavaBeans para servicios de entidad de negocio

El extremo Enterprise JavaBeans (EJB) es el extremo que subyace para todos los tipos de llamadas de servicios de entidad de negocio. Todos los demás extremos se asignan al extremo EJB.

Los servicios de entidad de negocio se muestran como un EJB sin estado. Un contenedor EJB sin estado puede agrupar instancias, asignar instancias y aplicar estrategias de equilibrio de cargas para distribuir la carga en diferentes servidores dentro del dominio.

Un extremo EJB acepta un nombre de usuario y una contraseña para autenticación.

Extremo REST para servicios de entidad de negocio

Con las llamadas de extremo de transferencia de estado representacional (REST), todos los servicios de entidad de negocio estarán disponibles como servicios web.

Los archivos Web Application Description Language (WADL) contienen descripciones XML de los servicios web REST, todas las direcciones URL de REST y todos los parámetros REST. MDM Hub genera un archivo WADL para cada Almacén de referencias operativas.

Puede descargar los archivos WADL para cada Almacén de referencias operativas desde la siguiente ubicación:

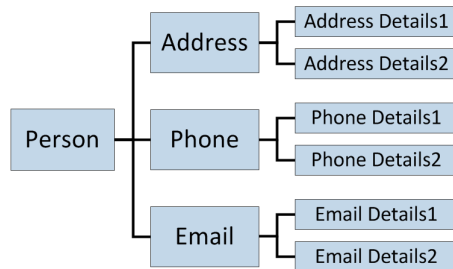
```
http://<host>:<puerto>/cmx/csfiles
```

Llamadas del servicio de entidad de negocio de REST y EJB

Cuando realice una llamada de servicio de entidad de negocio, puede especificar determinadas sucursales secundarias en lugar de solicitar la entidad de negocio completa.

Por ejemplo, desea realizar una operación de lectura en una entidad empresarial que tenga un nodo raíz de persona y varias sucursales secundarias. El objeto base de persona contiene los objetos base secundarios de dirección, teléfono, y correo electrónico. Cada objeto base secundario tiene dos objetos base secundarios de segundo nivel.

La siguiente imagen muestra la estructura de una entidad empresarial con varias sucursales:



Puede leer desde varias sucursales secundarias situadas en distintos niveles en una única solicitud. Por ejemplo, puede leer la persona, el teléfono, los detalles de teléfono 1, los detalles de teléfono 2, el correo electrónico y los detalles del correo electrónico 2 con una única solicitud.

El siguiente ejemplo de URL muestra cómo es posible hacer que una solicitud de lectura de REST obtenga el registro Persona con el ID de fila 1242, además de los registros secundarios Detalles de dirección 1 y Correo electrónico:

```
http://localhost:8080/cmx/cs/localhost-ORCL-DS_UI1/Person/1242?children=Address/  
Address_Details_1,Email
```

Extremo SOAP para servicios de entidad de negocio

Las llamadas de extremo de Protocolo simple de acceso a objetos (SOAP) hacen que los servicios de entidad de negocio estén disponibles como servicios web.

Los archivos de lenguaje de descripción de servicios web (WSDL) contienen las descripciones XML de los servicios web, los formatos de las respuestas y las solicitudes SOAP y todos los parámetros. MDM Hub genera un archivo WSDL para cada Almacén de referencias operativas.

Identificar un registro raíz

Puede usar uno de los siguientes métodos para identificar un registro raíz:

- rowid. Valor de la columna ROWID_OBJECT del registro.
- systemName y sourceKey. systemName es el nombre del sistema al que pertenece el registro. sourceKey es el valor de la columna PKEY_SRC_OBJECT del registro.
- Identificador de negocio global (GBID) de un objeto. Un GBID puede ser un valor compuesto, en cuyo caso debe pasar todos los valores.

Nota: El método GBID funciona únicamente con el servicio ReadBE.

En el siguiente código de ejemplo se usan systemName y sourceKey para identificar un registro:

```
String systemName = "SFA";  
  
Properties config = new Properties();  
config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL, EjbSiperianClient.PROTOCOL_NAME);  
CompositeServiceClient client = CompositeServiceClient.newCompositeServiceClient(config);  
CallContext callContext = new CallContext(orsId, user, pass);  
helperContext = client.getHelperContext(callContext);  
DataFactory dataFactory = helperContext.getDataFactory();  
  
//String personRowId = "1097";  
String pkeySrcObject = "CST1379";
```

```
//Set custom key pkey
pkey = (Key) dataFactory.create(Key.class);
pkey.setSystemName(systemName);
pkey.setSourceKey(val);
writePerson.setKey(pkey);
```

Filtros de datos y seguridad

Cuando los objetos base y los recursos tienen privilegios de funciones de usuario, la entidad de negocio hereda esos privilegios. Para acceder a los registros de entidad de negocio, la función de usuario debe tener los privilegios necesarios para el objeto base raíz de la entidad de negocio, así como de otros recursos.

Los servicios de entidad de negocio también heredan los filtros de datos que se establecen en los campos de la entidad de negocio.

Si desea más información sobre los filtros de datos y la seguridad, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

CAPÍTULO 2

Llamadas de servicio de entidad de negocio Enterprise Java Bean

Este capítulo incluye los siguientes temas:

- [Resumen de las llamadas de servicio de entidad de negocio Enterprise Java Bean, 14](#)
- [Ejemplo de código Java con clases SDO estándares, 14](#)
- [Ejemplo de código Java con clases SDO generadas, 18](#)

Resumen de las llamadas de servicio de entidad de negocio Enterprise Java Bean

Puede realizar llamadas de servicios de entidad de negocio Enterprise Java Bean (EJB) para crear, actualizar, eliminar y buscar registros de objetos base en una entidad de negocio. Puede crear código Java para ejecutar llamadas de servicios de entidad de negocio EJB.

Puede crear el código Java en función de las clases estándar de Objetos de servicio de datos (SDO) o bien puede crearlo en función de las clases java que genera MDM Hub según la configuración de la entidad de negocio y de los servicios de entidad de negocio.

Ejemplo de código Java con clases SDO estándares

En el ejemplo se muestra el código Java para ejecutar llamadas Enterprise Java Bean (EJB) basadas en las clases de Objeto de servicio de datos (SDO) estándar.

El ejemplo se encuentra en el siguiente archivo del kit de recursos: C:\<directorio de instalación de MDM Hub>\hub\resourcekit\samples\COS\source\java\com\informatica\mdm\sample\cs\DynamicSDO.java

El siguiente código Java se basa en las clases SDO estándar y ejecuta llamadas de servicio de entidad de negocio EJB para crear un registro de objeto base de persona, agregar varios registros secundarios, eliminar un registro secundario y, a continuación, eliminar el registro de persona y todos los registros secundarios:

```
package com.informatica.mdm.sample.cs;

import com.informatica.mdm.cs.CallContext;
import com.informatica.mdm.cs.api.CompositeServiceException;
import com.informatica.mdm.cs.client.CompositeServiceClient;
```

```

import com.siperian.sif.client.EjbSiperianClient;
import com.siperian.sif.client.SiperianClient;
import commonj.sdo.DataObject;
import commonj.sdo.Property;
import commonj.sdo.Type;
import commonj.sdo.helper.DataFactory;
import commonj.sdo.helper.HelperContext;

import java.io.PrintStream;
import java.util.Arrays;
import java.util.Properties;

public class DynamicSDO {

    public static void main(String[] args) throws CompositeServiceException {

        if(args.length != 3) {
            System.err.println("USAGE: DynamicSDO <ors> <user> <pass>");
            return;
        }

        new DynamicSDO(args[0], args[1], args[2]).execute();
    }

    private String orsId;
    private String user;
    private String pass;
    private HelperContext helperContext;
    private PrintStream out = System.out;

    public DynamicSDO(String orsId, String user, String pass) {
        this.orsId = orsId;
        this.user = user;
        this.pass = pass;
    }

    public void execute() throws CompositeServiceException {

        String systemName = "Admin";

        Properties config = new Properties();
        config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL,
EjbSiperianClient.PROTOCOL_NAME);
        CompositeServiceClient client =
CompositeServiceClient.newCompositeServiceClient(config);

        CallContext callContext = new CallContext(orsId, user, pass);

        helperContext = client.getHelperContext(callContext);

        DataFactory dataFactory = helperContext.getDataFactory();

        // types for Read requests
        Type coFilterType = helperContext.getTypeHelper().getType("urn:cs-
base.informatica.mdm", "CoFilter");
        Type coFilterNodeType = helperContext.getTypeHelper().getType("urn:cs-
base.informatica.mdm", "CoFilterNode");
        Type keyType = helperContext.getTypeHelper().getType("urn:cs-
base.informatica.mdm", "Key");

        // ReadCO & WriteCO request types
        Type readPersonType = helperContext.getTypeHelper().getType("urn:cs-
ors.informatica.mdm", "ReadPerson");
        Type writePersonType = helperContext.getTypeHelper().getType("urn:cs-
ors.informatica.mdm", "WritePerson");

        // 1. Create new person
        DataObject createPerson = dataFactory.create(writePersonType);
        DataObject createPersonParameters = createPerson.createDataObject("parameters");
        createPersonParameters.setString("systemName", systemName);
    }
}

```

```

DataObject person = createPerson.createDataObject("object");

person.getChangeSummary().beginLogging();

DataObject personRoot = person.createDataObject("Person");
personRoot.setString("firstName", "John");
personRoot.setString("lastName", "Smith");

person.getChangeSummary().endLogging();

dump("*** CREATE NEW PERSON ...", createPerson);

DataObject createPersonResponse = client.process(callContext, createPerson);

dump("*** PERSON CREATED:", createPersonResponse);

String personRowId = createPersonResponse.getString("object/Person/rowidObject");

DataObject readPerson = dataFactory.create(readPersonType);
DataObject readPersonParameters = readPerson.createDataObject("parameters");
DataObject coFilter = readPersonParameters.createDataObject("coFilter");
DataObject coFilterNode = coFilter.createDataObject("object");
coFilterNode.set("name", "Person");
DataObject key = coFilterNode.createDataObject("key");
key.set("rowid", personRowId);

dump("*** READ CREATED PERSON...", readPerson);

DataObject readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

personRoot = person.getDataObject("Person");
// add new 'one' child
DataObject genderCd = personRoot.createDataObject("genderCd");
genderCd.setString("genderCode", "M");

// add two 'many' children
DataObject phonePager = personRoot.createDataObject("TelephoneNumbers");
Property item = phonePager.getInstanceProperty("item");
Type phoneType = item.getType();

DataObject phone1 = dataFactory.create(phoneType);
phone1.setString("phoneNumber", "111-11-11");
DataObject phone2 = dataFactory.create(phoneType);
phone2.setString("phoneNumber", "222-22-22");

phonePager.setList(item, Arrays.asList(phone1, phone2));

person.getChangeSummary().endLogging();

DataObject updatePerson = dataFactory.create(writePersonType);
updatePerson.setDataObject("object", person);
DataObject updatePersonParameters = updatePerson.createDataObject("parameters");
updatePersonParameters.setString("systemName", systemName);
updatePersonParameters.setString("interactionId", "");

dump("*** UPDATE PERSON...", updatePerson);

DataObject updatePersonResponse = client.process(callContext, updatePerson);

dump("*** PERSON UPDATED:", updatePersonResponse);

coFilterNode.set("depth", 3);

readPersonParameters.setBoolean("readSystemFields", true);

```



```

dump("*** READ UPDATED PERSON WITH CHILDREN...", readPerson);

readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

genderCd = person.getDataObject("Person").createDataObject("genderCd");
genderCd.setString("genderCode", "F");

// delete one phone
DataObject phoneItem = person.getDataObject("Person/TelephoneNumbers/item[1]");
phoneItem.delete();

person.getChangeSummary().endLogging();

DataObject deletePhone = dataFactory.create(writePersonType);
deletePhone.setDataObject("object", person);
DataObject deletePhoneParameters = deletePhone.createDataObject("parameters");
deletePhoneParameters.setString("systemName", systemName);

dump("*** DELETE CHILD...", deletePhone);

DataObject deletePhoneResponse = client.process(callContext, deletePhone);

dump("*** CHILD DELETED:", deletePhoneResponse);

readPersonParameters.setBoolean("readSystemFields", false);

dump("*** READ PERSON AFTER CHILD WAS DELETED...", readPerson);

readPersonResponse = client.process(callContext, readPerson);

dump("*** READ RESULT:", readPersonResponse);

person = readPersonResponse.getDataObject("object");
person.detach();

person.getChangeSummary().beginLogging();

person.getDataObject("Person").detach();

person.getChangeSummary().endLogging();

DataObject deletePerson = dataFactory.create(writePersonType);
deletePerson.setDataObject("object", person);
DataObject deletePersonParameters = deletePerson.createDataObject("parameters");
deletePersonParameters.setString("systemName", systemName);

dump("*** DELETE PERSON...", deletePerson);

DataObject deletePersonResponse = client.process(callContext, deletePerson);

dump("*** PERSON DELETED:", deletePersonResponse);

dump("*** TRY TO READ PERSON AFTER DELETE", readPerson);

try {
    readPersonResponse = client.process(callContext, readPerson);

    dump("*** READ RESULT:", readPersonResponse);
} catch (CompositeServiceException e) {
    out.println("*** READ RESULT: " + e.getLocalizedMessage());
}

```

```

    }

    private void dump(String title, DataObject dataObject) {
        String xml = helperContext.getXMLHelper().save(
            dataObject,
            dataObject.getType().getURI(),
            dataObject.getType().getName());
        out.println(title);
        out.println(xml);
        out.println();
    }
}
}

```

Ejemplo de código Java con clases SDO generadas

En el ejemplo se muestra el código Java para ejecutar llamadas Enterprise Java Bean (EJB) basadas en las clases Java que genera MDM Hub en función de la configuración de la entidad de negocio y de los servicios de entidad de negocio.

El ejemplo se encuentra en el siguiente archivo del kit de recursos: C:\<directorio de instalación de MDM Hub>\hub\resourcekit\samples\COS\source\java\com\informatica\mdm\sample\cs\GeneratedSDO.java

El siguiente código Java se basa en las clases generadas y ejecuta llamadas de servicio de entidad de negocio EJB para crear un registro de objeto base de persona, agregar varios registros secundarios, eliminar un registro secundario y, después, eliminar el registro de persona y todos los registros secundarios:

```

package com.informatica.mdm.sample.cs;

import com.informatica.mdm.cs.CallContext;
import com.informatica.mdm.cs.api.CompositeServiceException;
import com.informatica.mdm.cs.client.CompositeServiceClient;
import com.informatica.mdm.sdo.cs.base.CoFilter;
import com.informatica.mdm.sdo.cs.base.CoFilterNode;
import com.informatica.mdm.sdo.cs.base.Key;
import com.siperian.sif.client.EjbsSiperianClient;
import com.siperian.sif.client.SiperianClient;
import commonj.sdo.DataObject;
import commonj.sdo.helper.DataFactory;
import commonj.sdo.helper.HelperContext;
import mdm.informatica.co_ors.*;
import mdm.informatica.cs_ors.*;

import java.io.PrintStream;
import java.util.Arrays;
import java.util.Properties;

public class GeneratedSDO {

    public static void main(String[] args) throws CompositeServiceException {

        if(args.length != 3) {
            System.err.println("USAGE: GeneratedSDO <ors> <user> <pass>");
            return;
        }

        new GeneratedSDO(args[0], args[1], args[2]).execute();
    }

    private String orsId;
    private String user;
    private String pass;
    private HelperContext helperContext;

```

```

private PrintStream out = System.out;

public GeneratedSDO(String orsId, String user, String pass) {
    this.orsId = orsId;
    this.user = user;
    this.pass = pass;
}

public void execute() throws CompositeServiceException {

    String systemName = "Admin";

    Properties config = new Properties();
    config.put(SiperianClient.SIPERIANCLIENT_PROTOCOL,
EjbSiperianClient.PROTOCOL_NAME);
    CompositeServiceClient client =
CompositeServiceClient.newCompositeServiceClient(config);

    CallContext callContext = new CallContext(orsId, user, pass);

    helperContext = client.getHelperContext(callContext);

    DataFactory dataFactory = helperContext.getDataFactory();

    // 1. Create new person
    WritePerson createPerson = (WritePerson)dataFactory.create(WritePerson.class);
    WritePersonParameters createPersonParameters =
(WritePersonParameters)dataFactory.create(WritePersonParameters.class);
    createPersonParameters.setSystemName(systemName);
    createPerson.setParameters(createPersonParameters);

    Person person = (Person)dataFactory.create(Person.class);
    createPerson.setObject(person);

    person.getChangeSummary().beginLogging();

    PersonRoot personRoot = (PersonRoot)dataFactory.create(PersonRoot.class);
    personRoot.setFirstName("John");
    personRoot.setLastName("Smith");
    person.setPerson(personRoot);

    person.getChangeSummary().endLogging();

    dump("*** CREATE NEW PERSON ...", createPerson);

    WritePersonReturn createPersonResponse =
(WritePersonReturn)client.process(callContext, (DataObject)createPerson);

    dump("*** PERSON CREATED:", createPersonResponse);

    String personRowId =
createPersonResponse.getObject().getPerson().getRowidObject();

    Key key = (Key)dataFactory.create(Key.class);
    key.setRowid(personRowId);
    CoFilterNode coFilterNode = (CoFilterNode)dataFactory.create(CoFilterNode.class);
    coFilterNode.setName(Person.class.getSimpleName());
    coFilterNode.setKey(key);
    CoFilter coFilter = (CoFilter)dataFactory.create(CoFilter.class);
    coFilter.setObject(coFilterNode);
    ReadPersonParameters readPersonParameters =
(ReadPersonParameters)dataFactory.create(ReadPersonParameters.class);
    readPersonParameters.setCoFilter(coFilter);

    ReadPerson readPerson = (ReadPerson)dataFactory.create(ReadPerson.class);
    readPerson.setParameters(readPersonParameters);

    dump("*** READ CREATED PERSON...", readPerson);

    ReadPersonReturn readPersonResponse =
(ReadPersonReturn)client.process(callContext, (DataObject)readPerson);

```

```

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();
        ((DataObject)person).detach();

        person.getChangeSummary().beginLogging();

        personRoot = person.getPerson();
        // add new 'one' child
        LUGenderLookup genderCd =
(LUGenderLookup) dataFactory.create(LUGenderLookup.class);
        genderCd.setGenderCode("M");
        personRoot.setGenderCd(genderCd);

        // add two 'many' children
        PersonTelephoneNumbersPager phonePager =
(PersonTelephoneNumbersPager) dataFactory.create(PersonTelephoneNumbersPager.class);

        PersonTelephoneNumbers phone1 =
(PersonTelephoneNumbers) dataFactory.create(PersonTelephoneNumbers.class);
        phone1.setPhoneNumber("111-11-11");
        PersonTelephoneNumbers phone2 =
(PersonTelephoneNumbers) dataFactory.create(PersonTelephoneNumbers.class);
        phone2.setPhoneNumber("222-22-22");

        phonePager.setItem(Arrays.asList(phone1, phone2));
        personRoot.setTelephoneNumbers(phonePager);

        person.getChangeSummary().endLogging();

        WritePerson updatePerson = (WritePerson) dataFactory.create(WritePerson.class);
        updatePerson.setObject(person);
        WritePersonParameters updatePersonParameters =
(WritePersonParameters) dataFactory.create(WritePersonParameters.class);
        updatePersonParameters.setSystemName(systemName);
        updatePersonParameters.setInteractionId("");
        updatePerson.setParameters(updatePersonParameters);

        dump("*** UPDATE PERSON...", updatePerson);

        WritePersonReturn updatePersonResponse =
(WritePersonReturn) client.process(callContext, (DataObject)updatePerson);

        dump("*** PERSON UPDATED:", updatePersonResponse);

        coFilterNode.setDepth(3);

        readPersonParameters.setReadSystemFields(true);

        dump("*** READ UPDATED PERSON WITH CHILDREN (with system fields)...",
readPerson);

        readPersonResponse = (ReadPersonReturn) client.process(callContext,
(DataObject)readPerson);

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();
        ((DataObject)person).detach();

        person.getChangeSummary().beginLogging();

        // delete one phone
        person.getPerson().getTelephoneNumbers().getItem().remove(0);

        // change gender
        genderCd = (LUGenderLookup) dataFactory.create(LUGenderLookup.class);
        genderCd.setGenderCode("F");
        personRoot.setGenderCd(genderCd);

```

```

        person.getChangeSummary().endLogging();

        WritePerson deletePhone = (WritePerson) dataFactory.create(WritePerson.class);
        deletePhone.setObject(person);
        WritePersonParameters deletePhoneParameters =
        (WritePersonParameters) dataFactory.create(WritePersonParameters.class);
        deletePhoneParameters.setSystemName(systemName);
        deletePhone.setParameters(deletePhoneParameters);

        dump("*** DELETE CHILD...", deletePhone);

        WritePersonReturn deletePhoneResponse =
        (WritePersonReturn) client.process(callContext, (DataObject) deletePhone);

        dump("*** CHILD DELETED:", deletePhoneResponse);

        readPersonParameters.setReadSystemFields(false);

        dump("*** READ PERSON AFTER CHILD WAS DELETED...", readPerson);

        readPersonResponse = (ReadPersonReturn) client.process(callContext,
        (DataObject) readPerson);

        dump("*** READ RESULT:", readPersonResponse);

        person = readPersonResponse.getObject();
        ((DataObject) person).detach();

        person.getChangeSummary().beginLogging();

        ((DataObject) person).getPerson().delete();

        person.getChangeSummary().endLogging();

        WritePerson deletePerson = (WritePerson) dataFactory.create(WritePerson.class);
        deletePerson.setObject(person);
        WritePersonParameters deletePersonParameters =
        (WritePersonParameters) dataFactory.create(WritePersonParameters.class);
        deletePersonParameters.setSystemName(systemName);
        deletePerson.setParameters(deletePersonParameters);

        dump("*** DELETE PERSON...", deletePerson);

        WritePersonReturn deletePersonResponse =
        (WritePersonReturn) client.process(callContext, (DataObject) deletePerson);

        dump("*** PERSON DELETED:", deletePersonResponse);

        dump("*** TRY TO READ PERSON AFTER DELETE", readPerson);

        try {
            readPersonResponse = (ReadPersonReturn) client.process(callContext,
            (DataObject) readPerson);

            dump("*** READ RESULT:", readPersonResponse);
        } catch (CompositeServiceException e) {
            out.println("*** READ RESULT: " + e.getLocalizedMessage());
        }
    }

    private void dump(String title, Object object) {
        DataObject dataObject = (DataObject) object;
        String xml = helperContext.getXMLHelper().save(
            dataObject,
            dataObject.getType().getURI(),
            dataObject.getType().getName());
        out.println(title);
        out.println(xml);
        out.println();
    }
}

```

```
} }
```

CAPÍTULO 3

Llamadas del servicio de entidad de negocio de transferencia de estado representacional

Este capítulo incluye los siguientes temas:

- [Resumen de las API de REST para servicios de entidad de negocio, 23](#)
- [Métodos REST compatibles, 24](#)
- [Método de autenticación, 24](#)
- [Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros, 24](#)
- [Archivos de idioma de descripción de aplicación web, 25](#)
- [Localizador uniforme de recursos de REST, 26](#)
- [Configuración de encabezado y cuerpo, 27](#)
- [Parámetros de consulta estándar, 29](#)
- [Formatos de fecha y hora UTC, 30](#)
- [Configuración de WebLogic para ejecutar llamadas REST de servicios de entidad de negocio, 31](#)
- [Visualización de parámetros de entrada y salida, 31](#)
- [Plantilla de JavaScript, 32](#)
- [Ejemplo de JavaScript, 33](#)
- [Referencia de API de REST para servicios de entidad de negocio , 35](#)

Resumen de las API de REST para servicios de entidad de negocio

Con las llamadas de extremo REST, todos los servicios de entidad de negocio estarán disponibles como servicios web.

Puede realizar llamadas de REST para crear, actualizar, eliminar y buscar registros de objeto base y registros secundarios relacionados en una entidad de negocio. Puede realizar operaciones como fusionar, anular la fusión y hacer coincidir registros. Puede realizar llamadas de REST para crear, actualizar y buscar tareas, así como llevarlas a cabo. También puede realizar llamadas de REST para crear, actualizar y eliminar archivos, como adjuntos para tareas o registros.

Una llamada de servicio de entidad de negocio de REST es una solicitud de servicio web con la forma de un localizador uniforme de recursos (URL). MDM Hub asigna una única URL a cada objeto base de una entidad de negocio. Puede utilizar la URL única para identificar el objeto base que desea actualizar o eliminar.

Nota: Antes de que utilice las API de REST para llamar a los servicios de entidad de negocio, valide el Almacén de referencias operativas.

Métodos REST compatibles

Las API de REST para servicios de entidad de negocio utilizan los métodos HTTP estándar para realizar operaciones en los recursos, como son los registros, las tareas y los archivos.

Las API de REST para servicios de entidad de negocio admiten los siguientes métodos de solicitud HTTP:

Método	Descripción
GET	Recupera información sobre un registro, una tarea o un archivo.
POST	Crea una tarea, un registro raíz, un registro secundario o un archivo. Nota: El nombre del Almacén de referencias operativas (ORS) es una solicitud POST con distinción de mayúsculas y minúsculas. Si el nombre del ORS no coincide con el nombre en MDM Hub, se produce un error.
PUT	Actualiza un registro raíz, un registro secundario, una tarea o un archivo.
PATCH	Actualiza una tarea parcialmente.
DELETE	Elimina un registro raíz, un registro secundario o un archivo.

Método de autenticación

Los extremos REST de los servicios de entidad de negocio utilizan el método de autenticación HTTP básico para autenticar usuarios. Cuando se conecte por primera vez a un servicio de entidad de negocio con su navegador, tendrá que proporcionar el nombre de usuario y la contraseña de MDM Hub. Cuando se haya autenticado correctamente, podrá utilizar las API de REST para servicios de entidad de negocio para realizar operaciones.

El navegador almacena en la caché las credenciales de usuario y las emplea en cada solicitud para acceder a un servicio de entidad de negocio que se produzca con posterioridad.

Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros

Utilice las cookies de autenticación para autenticar los usuarios de MDM Hub y llamar a los servicios de entidades de negocio de aplicaciones de terceros. Puede obtener una cookie basada en las credenciales de

un usuario autenticado. Guarde la cookie y utilícela para llamar a la API de REST. No es necesario que integre el nombre de usuario y la contraseña como parte del código.

Realice la siguiente solicitud de POST e inicie sesión en la vista Entidad 360 con su nombre de usuario y contraseña:

```
POST http://<host>:<port>/e360/com.informatica.tools.mdm.web.auth/login
{
  user: 'admin',
  password: 'user password'
}
```

Cuando la operación de inicio de sesión se realiza correctamente, el servidor devuelve la cookie de autenticación en el campo de encabezado `set-cookie`. El siguiente código de ejemplo muestra `set-cookie` en el encabezado de respuesta:

```
Set-Cookie: auth_hash_cookie="admin===QTc1RkNGQkNCMzc1RjIyOQ==";
Version=1; Path=/
```

Almacene el código hash y utilícelo en el encabezado de la solicitud de sus llamadas de API. No es necesario que proporcione un nombre de usuario y una contraseña para las llamadas de API.

El siguiente ejemplo muestra cómo se utiliza la cookie de autenticación en el encabezado de solicitud de la API:

```
GET http://<IP of host>:8080/cmxc/cs/localhost-orcl-DS_UI1/Person?action=meta
Cookie: auth_hash_cookie="admin===QTc1RkNGQkNCMzc1RjIyOQ=="
```

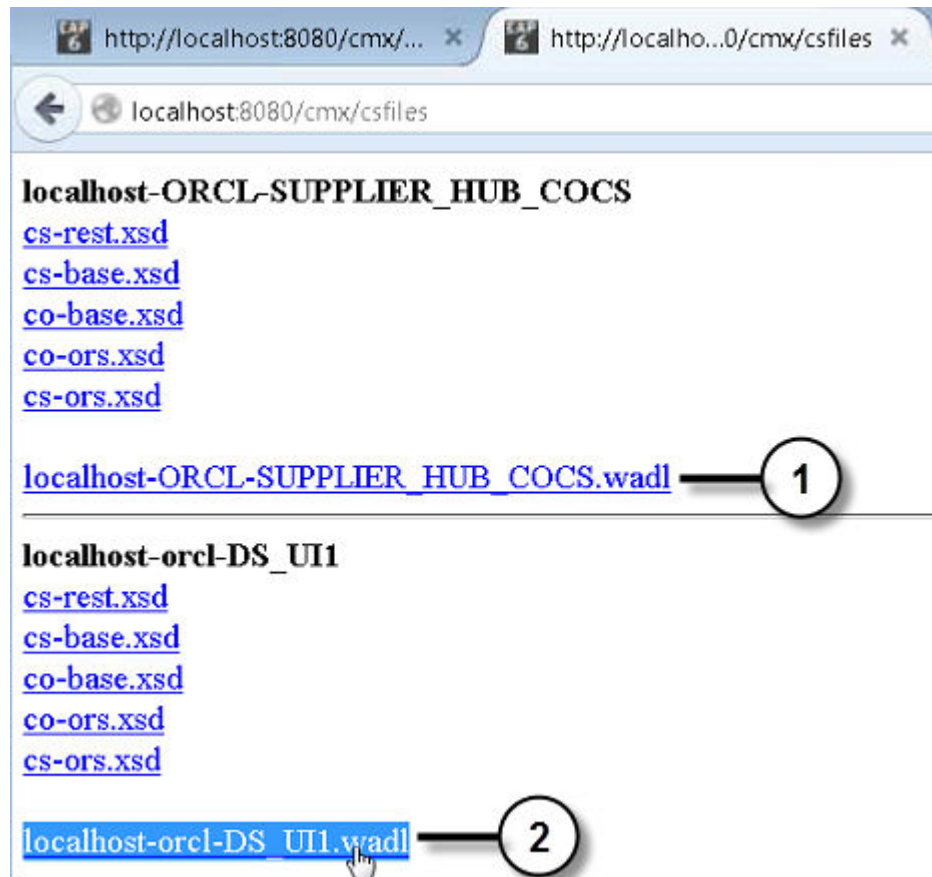
Archivos de idioma de descripción de aplicación web

Los archivos de idioma de descripción de aplicación web (WADL) contienen descripciones XML de los servicios web REST, todas las direcciones URL de REST y todos los parámetros REST. MDM Hub genera un archivo WADL para cada Almacén de referencias operativas.

Los archivos WADL para cada Almacén de referencias operativas se encuentran en la siguiente ubicación:

```
http://<host>:<puerto>/cmxc/csfiles
```

La siguiente imagen muestra la ubicación en la que puede descargar el archivo WADL para los almacenes de referencias operativas:



1. Enlace para descargar el archivo WADL para el Almacén de referencias operativas SUPPLIER_HUB_COCS
2. Enlace para descargar el archivo WADL para el Almacén de referencias operativas DS_UI1

Localizador uniforme de recursos de REST

Utilice una URL de REST para realizar llamadas REST para servicios de entidad de negocio.

La URL de REST tiene la siguiente sintaxis:

```
http://<host>:<puerto>/<contexto>/<ID de base de datos>/<ruta de acceso>
```

La URL tiene los siguientes campos:

host

El host que está ejecutando la base de datos.

puerto

Número de puerto que utiliza el servicio de escucha de la base de datos.

contexto

El contexto para las API de entidad de negocio y las API de tarea es `cmx/cs`.

El contexto para las API de archivo es `cmx/file`.

ID de base de datos

El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.

ruta de acceso

El objeto sobre el que quiere usar la API, como registros, tareas o archivos.

Si la dirección URL corresponde a un registro raíz, la ruta de acceso será el nombre del objeto raíz seguido de un identificador único.

Un ejemplo de una ruta de acceso para un registro raíz de persona puede ser `Person/798243.json`.

Si la dirección URL corresponde a un registro que sea un elemento secundario directo del objeto raíz, la ruta también incluiría el nombre del registro secundario y un identificador único.

Un ejemplo de una ruta para un registro de dirección de facturación que sea un elemento secundario de un registro raíz de persona sería:

```
Person/798243/BillAddresses/121522.json.
```

Si la dirección URL es para un registro secundario que esté a un nivel de dos o más, la ruta también incluiría la profundidad.

La siguiente URL es un ejemplo de una URL de REST para un registro secundario con una profundidad de 2:

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/798243/BillAddresses/121522.json?depth=2
```

Nota: Los parámetros distinguen entre mayúsculas y minúsculas. Asegúrese de que el caso de los nombres de parámetro de la URL de REST coincida con el caso de los nombres de parámetro de la configuración de REST.

Configuración de encabezado y cuerpo

Una operación REST combina un método HTTP con la dirección URL completa al recurso. Para obtener una solicitud completa, combine la operación REST con los encabezados HTTP correspondientes y otros datos necesarios. Una solicitud REST posee un componente de encabezado y de cuerpo. Para definir una solicitud se puede utilizar el formato JSON o XML.

Encabezado de solicitud

Utilice un encabezado de solicitud para definir los parámetros operativos o los metadatos de la operación REST. El encabezado consta de una serie de pares campo-valor. La línea de solicitud de la API contiene el método y la dirección URL. Especifique los campos del encabezado después de la línea de la solicitud.

Para crear el encabezado de solicitud de la API de REST, agregue los campos de encabezado tras la línea de solicitud `<METHOD> <<host>:<port>/<context>/<database ID>/<Path>`, como se muestra en el siguiente ejemplo:

```
<METHOD> <<host>:<port>/<context>/<database ID>/<Path>
Content-Type: application/<json/xml>
Accept: application/<json/xml>
```

En la siguiente tabla se describen algunos de los campos de encabezado de solicitud más frecuentes:

Componente de solicitud	Descripción
Content-Type	El tipo de medio de los datos en la solicitud. Al incluir un cuerpo en la solicitud de REST, debe especificar el tipo de medio del cuerpo en el campo de encabezado Content-Type. Incluya el campo de encabezado Content-Type en las solicitudes PUT y POST.
Accept	El tipo de medio de los datos de la respuesta. Para especificar el formato de la solicitud, utilice <code>application/<json/xml></code> en el encabezado o agregue <code>.json</code> o <code>.xml</code> a la URL. El valor predeterminado es XML.

Cuerpo de la solicitud

Utilice el cuerpo de solicitud de la API de REST para enviar datos en la solicitud. El cuerpo de la solicitud se utiliza con un método en el que se puede anexar un cuerpo, como los métodos POST y PUT. El cuerpo de los datos se escribe después de las líneas de encabezado. Si el mensaje de la solicitud incluye un cuerpo, utilice el campo de encabezado Content-Type para especificar el formato del cuerpo en el encabezado de la solicitud.

Los archivos de definición de esquema XML (XSD) describen qué elementos y atributos se pueden utilizar. El contenido del cuerpo de la solicitud depende de los tipos de elementos que se definan en los archivos XSD.

Los archivos XSD se encuentran en la siguiente ubicación:

```
http://<host>:<puerto>/cmx/csfiles
```

Formato XML

Cuando utilice el formato de solicitud XML, defina el objeto de la solicitud como un conjunto delimitador de etiquetas.

Utilice el siguiente formato XML para definir un objeto de solicitud:

```
<request object>
  <attribute1>value1</attribute1>
  <attribute2>value2</attribute2>
</request object>
```

En el siguiente ejemplo se muestra la representación XML de un objeto de solicitud:

```
<task>
  <taskType>
    <name>UpdateWithApprovalWorkflow</name>
  </taskType>
  <taskId>urn:b4p2:5149</taskId>
  <owner>manager</owner>
  <title>Smoke test task 222</title>
  <comments>Smoke testing</comments>
  <dueDate>2015-06-15T00:00:00</dueDate>
  <status>OPEN</status>
  <priority>NORMAL</priority>
  <creator>admin</creator>
  <createDate>2015-06-15T00:00:00</createDate>
  <updatedBy>admin</updatedBy>
  <lastUpdateDate>2015-06-15T00:00:00</lastUpdateDate>
  <businessEntity>Person</businessEntity>
  <orsId>localhost-orcl-DS_UI1</orsId>
  <processId>IDDUpdateWithApprovalTask</processId>
  <taskRecord>
    <businessEntity>
      <name>Person</name>
```

```

        <key>
            <rowid>123</rowid>
            <systemName></systemName>
            <sourceKey></sourceKey>
        </key>
    </businessEntity>
</taskRecord>
</task>

```

Formato JSON

Cuando utilice el formato de solicitud JSON, defina el objeto de solicitud con el atributo de tipo.

Utilice el siguiente formato JSON para especificar un objeto de solicitud:

```

{
  "type"="<request object>"
  "<attribute1>": "<value1>",
  "<attribute2>": "<value2>",
}

```

En el siguiente ejemplo se muestra la representación de JSON de un objeto de solicitud:

```

{
  "type"="task"
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "urn:b4p2:5149",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: 'IDDUpdateWithApprovalTask',
  taskRecord: [{
    businessEntity:{
      name: 'Person',
      key:{
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  ]
}

```

Parámetros de consulta estándar

Las API de REST de servicios de entidad de negocio utilizan parámetros de consulta estándar para filtrar, paginar y ampliar los resultados.

Utilice un signo de interrogación (?) para separar los parámetros de consulta del resto de parámetros. Los parámetros de consulta son pares clave-valor separados por el signo igual. Utilice un ampersand (&) para separar una secuencia de parámetros de consulta.

La siguiente URL de solicitud REST muestra cómo se deben usar los parámetros de consulta:

`/Person/123/Phone/SFA:456/PhoneUse?recordsToReturn=100&recordStates=ACTIVE,PENDING`

Utilice los siguientes parámetros de consulta estándar:

Parámetro	Descripción
recordsToReturn	Especifica el número de filas que se van a devolver. El valor predeterminado es 10.
firstRecord	Especifica la primera fila del resultado. El valor predeterminado es 1. Se utiliza en las llamadas posteriores para leer más páginas.
searchToken	Especifica el token de búsqueda que se devuelve con la solicitud anterior. Puede utilizar el token de búsqueda para obtener las páginas posteriores de los resultados de búsqueda. Por ejemplo, la siguiente consulta enumera la primera página: <code>/Person/123/Phone</code> La siguiente consulta devuelve la segunda página: <code>/Person/123/Phone?searchToken=SVR1.AZAM5&firstRecord=10</code>
returnTotal	Si se establece en true, devuelve el número de registros en el resultado. El valor predeterminado es false.
depth	Especifica el número de niveles de elementos secundarios que se van a incluir en el resultado.

Formatos de fecha y hora UTC

En la solicitud y la respuesta, todas las fechas y horas se especifican en formato UTC (hora universal coordinada) con o sin ajuste para una zona horaria específica.

Cuando especifique una fecha y una hora en el cuerpo de una solicitud, use uno de los formatos definidos en [Date and Time Formats \(NOTE-datetime\)](#) de acuerdo con la norma ISO 8601.

Las siguientes directrices se han extraído del documento NOTE-datetime:

Tipo	Sintaxis	Ejemplo
Fecha: año	YYYY	1997
Fecha: año y mes	YYYY-MM	1997-07
Fecha: año, mes y día	YYYY-MM-DD	1997-07-16
Fecha más horas y minutos	YYYY-MM-DDThh:mmTZD	1997-07-16T19:20+01:00
Fecha más horas, minutos y segundos	YYYY-MM-DDThh:mm:ssTZD	1997-07-16T19:20:30+01:00
Fecha más horas, minutos, segundos y segundos fraccionarios	YYYY-MM-DDThh:mm:ss.sTZD	1997-07-16T19:20:30.45+01:00

Donde:

- YYYY = un año de cuatro dígitos
- MM = un mes de dos dígitos, de 01 a 12
- DD = un día del mes de dos dígitos, de 01 a 31
- T = un valor literal que sigue a la fecha y presenta la hora
- hh = dos dígitos para la hora, de 00 a 23
- mm = dos dígitos para los minutos, de 00 a 59
- ss = dos dígitos para los segundos, de 00 a 59
- s = uno o varios dígitos que representan una fracción decimal de un segundo
- TZD = designador de zona horaria (Z o +hh:mm o -hh:mm)
 - Z para hora UTC
 - +hh:mm para una zona horaria local anterior a UTC
 - -hh:mm para una zona horaria local posterior a UTC

Configuración de WebLogic para ejecutar llamadas REST de servicios de entidad de negocio

Dado que las llamadas REST de servicios de entidad de negocio utilizan autenticación HTTP básica, debe deshabilitar la autenticación del Servidor WebLogic para las llamadas REST. Para configurar WebLogic de manera que ejecute las llamadas REST de servicios de entidad de negocio, edite el archivo `config.xml` de WebLogic.

1. Desplácese hasta el siguiente directorio de WebLogic:

En UNIX.

```
<directorio de instalación de WebLogic>/user_projects/domains/base_domain/config
```

En Windows.

```
<directorio de instalación de WebLogic>\user_projects\domains\base_domain\config
```

2. Abra el siguiente archivo en un editor de texto:

```
config.xml
```

3. Antes de la etiqueta `</security-configuration>` de cierre, agregue el siguiente código XML:

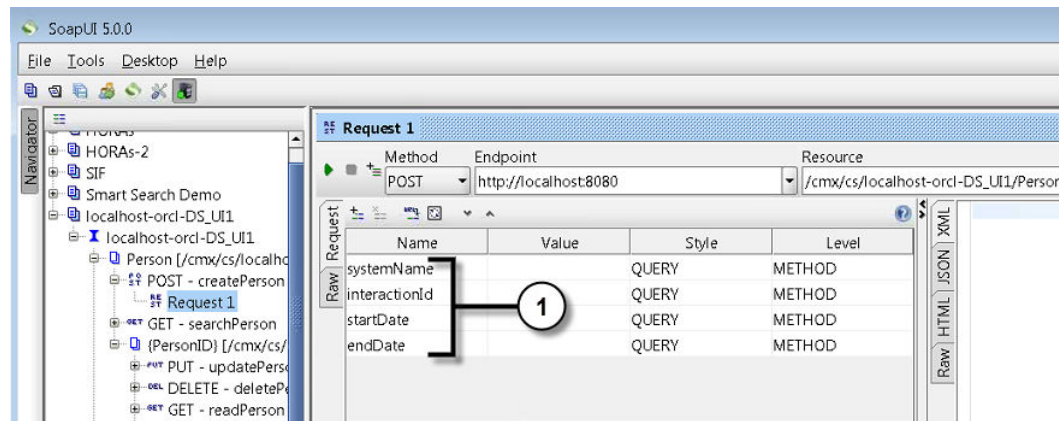
```
<enforce-valid-basic-auth-credentials>  
  false  
</enforce-valid-basic-auth-credentials>
```

Visualización de parámetros de entrada y salida

Puede utilizar una herramienta de pruebas funcional, como SoapUI, para ver los parámetros de entrada y salida de la API de REST.

Descargue el archivo WADL y, después, importe el archivo en la herramienta de pruebas funcional para crear un proyecto de REST.

La siguiente imagen muestra los parámetros de entrada en SoapUI para la API de REST createPerson:



1. Parámetros de entrada para la API de REST createPerson

Nota: Se podrían producir errores en la importación del archivo WADL generado en entornos de WebSphere al importarse en SoapUI. Si los parámetros de entrada no aparecen en SoapUI, edite el archivo WADL para quitar el atributo `xmlns` de cada elemento `param` y, a continuación, importar el archivo WADL nuevamente.

Plantilla de JavaScript

En el siguiente ejemplo de código se muestra una plantilla básica que se puede modificar para crear código JavaScript para las llamadas de servicio de entidad de negocio REST. Necesita la biblioteca de scripts `jquery`.

```
(function ($) {
    window.CSClient = window.CSClient || {
        baseUrl: "/cmx/cs/" + "[siperian-client.orsId]",
        user: "[siperian-client.username]",
        pass: "[siperian-client.password]",

        process: function (method, url, body, params) {
            var fullUrl = this.baseUrl + url + ".json?" + $.param(params);
            return $.ajax({
                method: method,
                contentType: "application/json",
                url: fullUrl,
                data: JSON.stringify(body),
                beforeSend: function (xhr) {
                    xhr.setRequestHeader("Authorization", "Basic " + btoa(CSClient.user
+ ":" + CSClient.pass));
                }
            });
        },

        readCo: function (url, params) {
            return this.process("GET", url, null, params);
        },
        createCo: function (url, body, params) {
            return this.process("POST", url, body, params);
        },
        updateCo: function (url, body, params) {
            return this.process("PUT", url, body, params);
        },
        deleteCo: function (url, params) {
            return this.process("DELETE", url, null, params);
        }
    }
})
```



```
    };  
  })(jQuery);
```

Ejemplo de JavaScript

El kit de recursos contiene ejemplos del código fuente de Java que muestran cómo realizar llamadas de servicio de entidad de negocio REST.

El código de ejemplo se muestra en el siguiente archivo:

```
<directorio de instalación de MDM Hub>\hub\resourcekit\samples\COS\source\resources\webapp  
\rest-api.html
```

El siguiente código muestra las llamadas de las API de REST para crear un registro de raíz de persona, agregar varios registros secundarios, eliminar un registro secundario y, a continuación, eliminar el registro de persona y todos los registros secundarios:

```
<html>  
<head>  
  <script type="text/javascript" src="jquery-1.11.1.js"></script>  
  <script type="text/javascript" src="cs-client.js"></script>  
</head>  
<body>  
  
<script type="text/javascript" language="javascript">  
  $(document).ready(function () {  
  
    $("#run").click(function () {  
  
      log = function(msg, json) {  
        $('#log').before("<hr/><b>" + msg + "</b>");  
        $('#log').before("<pre>" + JSON.stringify(json, undefined, 2) + "</  
pre>");  
      };  
  
      CSCClient.createCo(  
        "/Person",  
        {  
          firstName: "John",  
          lastName: "Smith"  
        },  
        {  
          systemName: "Admin"  
        }  
      ).then(  
        function (result) {  
          log("PERSON CREATED:", result);  
          return CSCClient.readCo(  
            "/Person/" + result.Person.rowidObject.trim(),  
            {  
              depth: 1  
            }  
          );  
        }  
      ).then(  
        function (result) {  
          log("READ CREATED PERSON:", result);  
          return CSCClient.updateCo(  
            "/Person/" + result.rowidObject.trim(),  
            {  
              genderCd: {  
                genderCode: "M"  
              },  
              TelephoneNumbers: {
```

```

                item: [
                    {
                        phoneNumber: "111-11-11"
                    },
                    {
                        phoneNumber: "222-22-22"
                    }
                ]
            },
            {
                systemName: "Admin"
            }
        );
    }
).then(
    function (result) {
        log("PERSON UPDATED:", result);
        return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
                depth: 3,
                readSystemFields: true
            }
        );
    }
).then(
    function (result) {
        log("READ UPDATED PERSON:", result);
        return CSClient.deleteCo(
            "/Person/" + result.rowidObject.trim() + "/"
            TelephoneNumbers/" + result.TelephoneNumbers.item[0].rowidObject.trim(),
            {
                systemName: "Admin"
            }
        );
    }
).then(
    function (result) {
        log("TELEPHONE DELETED:", result);
        return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
                depth: 3
            }
        );
    }
).then(
    function (result) {
        log("READ PERSON AFTER TELEPHONE IS DELETED:", result);
        return CSClient.deleteCo(
            "/Person/" + result.rowidObject.trim(),
            {
                systemName: "Admin"
            }
        );
    }
).then(
    function (result) {
        log("PERSON DELETED:", result);
        return CSClient.readCo(
            "/Person/" + result.Person.rowidObject.trim(),
            {
                depth: 1,
                recordStates: "ACTIVE,PENDING,DELETED",
                readSystemFields: true
            }
        );
    }
).then(
    function (result) {

```

```

        log("READ PERSON AFTER DELETE (HSI -1):", result);
    }
    });
});
</script>

<input type="button" id="run" value="Run..."/>
<p/>
<div id="log"></div>
</body>
</html>

```

Referencia de API de REST para servicios de entidad de negocio

La referencia de la API de REST para servicios de entidad de negocio enumera las API de REST y proporciona una descripción de cada una de ellas. La referencia de API también contiene información sobre las direcciones URL, los parámetros de consulta, las solicitudes de muestra y las respuestas de ejemplo.

Obtener metadatos

La API de REST de obtención de metadatos devuelve la estructura de datos de una entidad de negocio o una relación de entidad de negocio.

La API utiliza el método GET para devolver los siguientes metadatos de una entidad de negocio:

- Estructura de la entidad de negocio
- Lista de campos
- Tipos de campos, como el tipo y el tamaño de los datos
- Configuración de seguridad para operaciones, como crear, actualizar y fusionar
- Etiquetas localizadas para nodos o campos
- Nombre del código y campos de visualización para campos de búsqueda

La API devuelve los siguientes detalles de una relación de entidad de negocio:

- Nombre y etiqueta de la relación.
- Entidades de negocio de origen y destino.
- Dirección de la relación.

URL de solicitud

La dirección URL de obtención de metadatos tiene el siguiente formato para una entidad de negocio:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=meta
```

La dirección URL de obtención de metadatos tiene el siguiente formato para una relación:

```
http://<host>:<port>/<context>/<database ID>/<relationship>?action=meta
```

Utilice el parámetro de consulta "action=meta" para recuperar la información de metadatos. Asegúrese de especificar correctamente el nombre de la entidad de negocio.

Haga una solicitud GET de HTTP a la URL para la obtención de metadatos:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity/relationship>?
action=meta
```

Puede agregar encabezados HTTP a la solicitud.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera la información de metadatos para la entidad de negocio de persona y el nodo raíz:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
```

La siguiente solicitud de ejemplo incluye un encabezado para recuperar la información de metadatos para la entidad de negocio de persona en el formato JSON:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
Accept: application/json
```

En la siguiente solicitud de ejemplo se recupera la información de los metadatos para la relación HouseholdContainsMemberPerson:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductGroupProductGroup?action=meta
```

Ejemplo de respuesta de API

Las respuestas de ejemplo para una entidad y una relación contienen permisos de seguridad. La primera sección `operations` define los posibles permisos. La sección `object` enumera los permisos para toda la entidad de negocio o relación. La sección `fields` define los permisos en el nivel de campo.

El siguiente ejemplo muestra una estructura de datos parcial para la entidad de negocio de persona en el formato JSON.

```
{
  "operations": {
    "read": {
      "allowed": true
    },
    "search": {
      "allowed": true
    },
    "create": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeUpdate",
          "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
      }
    },
    "update": {
      "allowed": true,
      "task": {
        "template": {
          "title": "Review changes in {taskRecord[0].label}",
          "priority": "NORMAL",
          "dueDate": "2018-04-24T09:28:13.455-04:00",
          "taskType": "AVOSBeUpdate",
          "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
```

```

        "attachment": "OPTIONAL"
    }
},
"merge": {
    "allowed": true,
    "task": {
        "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:28:13.455-04:00",
            "taskType": "AVOSBeMerge",
            "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
    }
},
"delete": {
    "allowed": true
},
"unmerge": {
    "allowed": true,
    "task": {
        "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:28:13.455-04:00",
            "taskType": "AVOSBeUnmerge",
            "comment": "This is urgent. Please review ASAP"
        },
        "comment": "AS_REQUIRED",
        "attachment": "OPTIONAL"
    }
}
},
"objectType": "ENTITY",
"timeline": true,
"object": {
    "operations": {
        "read": {
            "allowed": true
        },
        "create": {
            "allowed": true
        },
        "update": {
            "allowed": true
        },
        "merge": {
            "allowed": true
        },
        "delete": {
            "allowed": true
        },
        "unmerge": {
            "allowed": true
        }
    }
},
"field": [
    {
        "operations": {
            "read": {
                "allowed": true
            },
            "create": {
                "allowed": true
            },
            "update": {
                "allowed": true
            }
        }
    }
],

```

```

    "allowedValues": [
      "Person"
    ],
    "searchable": {
      "filterable": true,
      "facet": true
    },
    "name": "partyType",
    "label": "Party Type",
    "dataType": "String",
    "length": 255
  },
  {
    "operations": {
      "read": {
        "allowed": true
      },
      "create": {
        "allowed": true
      },
      "update": {
        "allowed": true
      }
    },
    "name": "lastName",
    "label": "Last Name",
    "dataType": "String",
    "length": 50
  },
  {
    "operations": {
      "read": {
        "allowed": true
      },
      "create": {
        "allowed": true
      },
      "update": {
        "allowed": true
      }
    },
    "searchable": {
      "filterable": true,
      "facet": true
    },
    "name": "displayName",
    "label": "Display Name",
    "dataType": "String",
    "length": 200
  },
  ...
],
"name": "Person",
"label": "Person",
"many": false
}
}

```

Lista de metadatos

La API de REST de lista de metadatos devuelve una lista de las entidades de negocio o las relaciones que ha definido. La respuesta incluye información de la línea temporal e información de seguridad, si las entidades de negocio contienen esa información. Puede usar la API para recuperar la lista de relaciones que empiezan en una entidad, finalizan en una entidad o empiezan y finalizan en las entidades especificadas.

La API usa el método GET.

Dirección URL de lista de metadatos

La dirección URL de lista de metadatos tiene el siguiente formato para los metadatos de entidad:

```
http://<host>:<port>/<context>/<database ID>/meta/entity
```

La dirección URL de lista de metadatos tiene el siguiente formato para los metadatos de relación:

```
http://<host>:<port>/<context>/<database ID>/meta/relationship
```

Use la siguiente solicitud HTTP GET para la dirección URL de lista de metadatos:

```
GET http://<host>:<port>/<context>/<database ID>/meta/entity|relationship
```

Parámetros de consulta

Puede anexar los parámetros de consulta a la dirección URL de la solicitud para filtrar los resultados de búsqueda para las relaciones de entidad de negocio. Puede especificar la dirección y la búsqueda de relaciones en y desde una entidad de negocio.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
start	Opcional. Especifica la entidad en la que se origina una relación. Por ejemplo, la consulta <code>meta/relationship?start=Organization</code> devuelve todas las relaciones que empiezan a partir de la entidad de negocio <code>Organization</code> .
finish	Opcional. Especifica la entidad en la que finaliza una relación. Por ejemplo, la consulta <code>meta/relationship?finish=Person</code> devuelve todas las relaciones que finalizan en la entidad de negocio <code>Person</code> .

Puede especificar ambos parámetros para recuperar todas las relaciones entre las dos entidades de negocio. Por ejemplo, la consulta `meta/relationship?start=Organization&finish=Person` devuelve todas las relaciones que empiezan por la entidad de negocio `Organization` y terminan por la entidad de negocio `Person`.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se recupera la lista de entidades de negocio configuradas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/entity
```

En la siguiente solicitud de ejemplo se recupera la lista de relaciones configuradas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/relationship
```

En la siguiente solicitud de ejemplo se recupera la lista de relaciones que se inician en la entidad de negocio `Organization` y finalizan en la entidad de negocio `Person`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/relationship?start=Organization&finish=Person
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra un fragmento de la lista de relaciones configuradas:

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
```

```

"item": [
  {
    "operations": {
      "read": {
        "allowed": true
      },
      "search": {
        "allowed": false
      },
      "create": {
        "allowed": true,
        "task": {
          "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeUpdate",
            "comment": "This is urgent. Please review ASAP"
          },
          "comment": "AS_REQUIRED",
          "attachment": "OPTIONAL"
        }
      },
      "update": {
        "allowed": true,
        "task": {
          "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeUpdate",
            "comment": "This is urgent. Please review ASAP"
          },
          "comment": "AS_REQUIRED",
          "attachment": "OPTIONAL"
        }
      },
      "merge": {
        "allowed": true,
        "task": {
          "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeMerge",
            "comment": "This is urgent. Please review ASAP"
          },
          "comment": "AS_REQUIRED",
          "attachment": "OPTIONAL"
        }
      },
      "delete": {
        "allowed": true
      },
      "unmerge": {
        "allowed": true,
        "task": {
          "template": {
            "title": "Review changes in {taskRecord[0].label}",
            "priority": "NORMAL",
            "dueDate": "2018-04-24T09:31:48.167-04:00",
            "taskType": "AVOSBeUnmerge",
            "comment": "This is urgent. Please review ASAP"
          },
          "comment": "AS_REQUIRED",
          "attachment": "OPTIONAL"
        }
      }
    },
    "objectType": "ENTITY",
    "timeline": false,
  }
]

```



```

"object": {
  "link": [
    {
      "href": "http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/
CreditCard.json?action=meta",
      "rel": "entity"
    }
  ],
  "field": [
    {
      "name": "issuingCompany",
      "label": "Issuing Company",
      "dataType": "String",
      "length": 100
    },
    {
      "name": "expirationYear",
      "label": "Expiration Year",
      "dataType": "String",
      "length": 4
    },
    {
      "allowedValues": [
        "Credit Card"
      ],
      "name": "accountType",
      "label": "Account Type",
      "dataType": "String",
      "length": 255
    },
    {
      "name": "accountNumber",
      "label": "Account Number",
      "dataType": "String",
      "length": 20
    },
    {
      "name": "securityCode",
      "label": "Security Code",
      "dataType": "String",
      "length": 4
    },
    {
      "name": "expirationMonth",
      "label": "Expiration Month",
      "dataType": "String",
      "length": 2
    },
    {
      "name": "cardholderName",
      "label": "Card Holder Name",
      "dataType": "String",
      "length": 100
    },
    {
      "name": "consolidationInd",
      "label": "Consolidation Ind",
      "dataType": "Integer",
      "length": 38,
      "readOnly": true,
      "system": true
    },
    {
      "name": "creator",
      "label": "Creator",
      "dataType": "String",
      "length": 50,
      "readOnly": true,
      "system": true
    }
  ]
}

```

```

    "name": "interactionId",
    "label": "Interaction Id",
    "dataType": "Integer",
    "length": 38,
    "readOnly": true,
    "system": true
  },
  {
    "name": "updatedBy",
    "label": "Updated By",
    "dataType": "String",
    "length": 50,
    "readOnly": true,
    "system": true
  },
  {
    "name": "lastUpdateDate",
    "label": "Last Update Date",
    "dataType": "Date",
    "readOnly": true,
    "system": true
  },
  {
    "name": "lastRowidSystem",
    "label": "Last Rowid System",
    "dataType": "String",
    "length": 14,
    "readOnly": true,
    "system": true
  },
  {
    "name": "dirtyIndicator",
    "label": "Dirty Indicator",
    "dataType": "Integer",
    "length": 38,
    "readOnly": true,
    "system": true
  },
  {
    "name": "deletedBy",
    "label": "Deleted By",
    "dataType": "String",
    "length": 50,
    "readOnly": true,
    "system": true
  },
  {
    "name": "deletedInd",
    "label": "Deleted Indicator",
    "dataType": "Integer",
    "length": 38,
    "readOnly": true,
    "system": true
  },
  {
    "name": "hubStateInd",
    "label": "Hub State Ind",
    "dataType": "Integer",
    "length": 38,
    "readOnly": true,
    "system": true
  },
  {
    "name": "deletedDate",
    "label": "Deleted Date",
    "dataType": "Date",
    "readOnly": true,
    "system": true
  },
  {
    "name": "rowidObject",

```

```

        "label": "Rowid Object",
        "dataType": "String",
        "length": 14,
        "readOnly": true,
        "system": true
    },
    {
        "name": "cmDirtyInd",
        "label": "Content metadata dirty Ind",
        "dataType": "Integer",
        "length": 38,
        "readOnly": true,
        "system": true
    },
    {
        "name": "createDate",
        "label": "Create Date",
        "dataType": "Date",
        "readOnly": true,
        "system": true
    }
],
"name": "CreditCard",
"label": "Credit Card",
"many": false
}
},
...
]
}

```

Leer registros

La API de REST de lectura de registros devuelve los detalles de un registro raíz en la entidad de negocio. Puede utilizar la API para devolver los detalles de los registros secundarios de un registro raíz. Igualmente, la API también se puede utilizar para ver los metadatos del contenido de un registro.

La API utiliza el método GET.

Puede ordenar el conjunto de resultados para ver la información en orden ascendente o descendente. Utilice el método POST cuando necesite parámetros complejos y en mayor cantidad. Por ejemplo, si desea recuperar los datos en los que los elementos secundarios se ordenan por un conjunto de campos.

URL de solicitud

Utilice el ID de fila o el nombre del sistema de origen y la clave de fuente para especificar el registro en la URL de solicitud.

La URL de lectura de registros puede tener los siguientes formatos:

URL con rowId

Utilice el siguiente formato de URL al especificar el ID de fila:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>
```

URL con nombre del sistema de origen y clave de fuente

Utilice el siguiente formato de URL al especificar el nombre del sistema de origen y la clave de fuente:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<system name>:<source key>
```

URL con el nombre de sistema y el identificador de negocio global (GBID) de un objeto

Utilice el siguiente formato de URL al especificar el nombre del sistema de origen y el GBID:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<system name>:uid:<gbid>
```

URL con solo el GBID

Utilice el siguiente formato de URL al especificar solo el GBID:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/:uid:<gbid>
```

URL con más de un GBID

Utilice el siguiente formato de URL al especificar más de un GBID:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/:one:<gbid>,another:<gbid>
```

URL para devolver los detalles de los nodos secundarios

Utilice el siguiente formato de URL para devolver los detalles de los nodos secundarios:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>?depth=n
```

URL para devolver los detalles de un nodo secundario

Utilice el siguiente formato de URL para devolver los detalles de los nodos secundarios:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>?children=<comma separated list of child node names or paths>
```

Por ejemplo, children= BillAddresses/Address,Email

URL para devolver los detalles de un nodo concreto

Utilice el siguiente formato de URL para devolver los detalles de un nodo concreto:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>/<node name>
```

URL para devolver los detalles de los elementos secundarios de un nodo concreto

Utilice el siguiente formato de URL para devolver los detalles de los elementos secundarios de un nodo concreto:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the record>/<node name>?children=<child node name>
```

URL para devolver los metadatos de contenido de un registro

Utilice el siguiente formato de URL para devolver los metadatos de contenido de un registro:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?contentMetadata=<content metadata type>
```

Por ejemplo, puede recuperar coincidencias para los registros secundarios con la siguiente solicitud GET:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?contentMetadata=MATCH
```

URL para ordenar los elementos secundarios por campos

Utilice el siguiente formato de URL para ordenar los elementos secundarios por campos:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>/<node name>?order=-<field name>
```

Utilice el carácter - como sufijo para especificar el orden descendente.

Parámetros de consulta

Puede anexar los parámetros de consulta a la URL de solicitud para filtrar los detalles del registro.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
depth	El número de niveles secundarios que se devolverán. Especifique 2 para devolver el nodo raíz y sus elementos secundarios directos, y 3 para devolver el nodo raíz, el elemento secundario directo y los elementos secundarios de segundo nivel. El valor predeterminado es 1.
effectiveDate	La fecha cuyos datos desea recuperar.
readSystemFields	Indica si desea devolver los campos del sistema en el resultado. El valor predeterminado es false.
recordStates	El estado del registro. Proporcione una lista de estados separados por comas. Los estados de registro admitidos son ACTIVE, PENDING y DELETED. El valor predeterminado es ACTIVE.
contentMetadata	Los metadatos del registro. Proporcione una lista de valores separados por comas. Por ejemplo, XREF, PENDING_XREF, DELETED_XREF, HISTORY, XREF_HISTORY y MATCH. Si selecciona MATCH, la respuesta contiene una lista de registros coincidentes recuperados de la tabla _MTCH.
historyDate	Fecha cuyos datos del historial desea recuperar. La respuesta contiene datos del registro para la fecha especificada que se han recuperado de la tabla _HIST. Puede usar historyDate con el parámetro contentMetadata para recuperar metadatos históricos. Establezca contentMetadata en XREF, BVT o TRUST. <ul style="list-style-type: none">- XREF. La respuesta contiene datos de referencias cruzadas históricos de la tabla _HXRF.- BVT. La respuesta contiene la mejor versión de confianza histórica de la tabla _HCTL.- TRUST. La respuesta contiene valores de configuración de confianza históricos de las tablas _HCTL y _HVXR.
children	Lista de los nombres de nodos secundarios o rutas separados por comas.

Parámetro	Descripción
suppressLinks	Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false. Por ejemplo, la consulta <code>Person/1242?depth=10&suppressLinks=true</code> mostrará los detalles del registro de hasta 10 niveles secundarios, sin incluir vínculos entre elementos primarios y secundarios en la respuesta.
order	Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo. Por ejemplo, la consulta <code>Person/1242/Names?order=-name</code> muestra el resultado con los nombres en orden descendente. La consulta <code>Person/1242/BillAddresses?order=rowidObject,-effStartDate</code> mostrará las direcciones de facturación con los valores de Rowid en orden ascendente y la fecha de inicio efectiva en orden descendente.

En el siguiente ejemplo se muestra cómo se deben filtrar los detalles de un registro:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123/Phone/SFA:456/PhoneUse?recordsToReturn=100&recordStates=ACTIVE,PENDING&contentMetadata=XREF
```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Solicitud POST para especificar el orden de clasificación de los elementos secundarios

Utilice una solicitud POST para ordenar el conjunto de resultados en varios campos. Incluya los parámetros o los campos en el cuerpo de la solicitud POST.

En la siguiente solicitud de ejemplo, se muestra cómo utilizar la solicitud POST para una operación de lectura y ordenar los datos en varios campos:

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/ReadPerson.json
{
  parameters:
  {
    coFilter: {
      object: {
        name: "Person",
        key: {
          rowid: 1242
        },
        order: "lastName",
        object: [
          {name: "Names", order: "-name"},
          {name: "Phone", order: "phoneNum, -phoneCountryCd",
            object: [{name: "PhonePermissions", order: "-column1"}]}
        ]
      }
    }
  }
}
```

Nota: En cada nivel de la entidad de negocio, solo se permite un tipo de orden de clasificación por cada tipo de elemento secundario.

Consideraciones del orden de clasificación

La API de lectura de registros es compatible con la ordenación de clasificación en uno o más campos para cada nodo secundario de entidad de negocio. La siguiente sección describe ciertas consideraciones que debe tomar en cuenta al especificar el orden de clasificación.

- Si especifica el orden de clasificación del elemento secundario de segundo nivel, pero no para el elemento secundario, el elemento secundario de segundo nivel se organiza en el orden especificado. El elemento secundario no se organiza en el orden de clasificación que ha especificado para el elemento secundario de segundo nivel. A continuación se muestra una solicitud de ejemplo:

```
http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/1242/Phone/861/PhonePermissions?order=-column1
```

En la solicitud de ejemplo, a pesar de que se ha especificado un orden de clasificación descendente para el elemento secundario de segundo nivel PhonePermissions, no se especifica ninguno para el elemento secundario Phone. Phone no se ha organizado en el orden de clasificación de PhonePermissions.

- Si especifica el orden de clasificación del elemento secundario y no el del elemento secundario de segundo nivel, el elemento secundario se organiza en el orden de clasificación que se ha especificado. El elemento secundario de segundo nivel no se organiza en el orden de clasificación que ha especificado para el elemento secundario. A continuación se muestra una solicitud de ejemplo:

```
{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", order:"-phoneCountryCd, -phoneNum", object:
          [{name:"PhonePermissions"}]},
        ]}
    }}}}
```

En la solicitud de ejemplo, se especifica el orden de clasificación para el elemento secundario Phone, pero no para el elemento secundario de segundo nivel PhonePermissions. El elemento secundario Phone se organiza en el orden que se ha especificado.

- Si especifica el orden de clasificación del elemento secundario y del elemento secundario de segundo nivel, ambos se organizan por el orden de clasificación. La siguiente solicitud de ejemplo especifica el orden de clasificación de Phone (elemento secundario) y de PhonePermissions (elemento secundario de segundo nivel):

```
{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", order:"-phoneCountryCd, -phoneNum", object:
          [{name:"PhonePermissions", order:"-column1"}]},
        ]}
    }}}}
```

- Un elemento secundario solo se puede ordenar por las columnas del propio elemento secundario, mientras que el elemento secundario de segundo nivel se puede ordenar por las columnas del elemento secundario de segundo nivel. En los siguientes ejemplos de solicitud, Phone se ordena por PhoneType y PhonePermissions se ordena por la columna 1. PhoneType es una columna de Phone (elemento secundario) y la columna 1 es una columna de PhonePermissions (elemento secundario de segundo nivel).

```
http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/1242/Phone?order=-PhoneType
http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/1242/Phone/861/phonePermissions?order=column1
```

- En cada nivel de la entidad de negocio, solo se permite un tipo de orden de clasificación por cada tipo de elemento secundario. En la siguiente solicitud, se han especificado diferentes tipos de órdenes de

clasificación para los elementos secundarios de PhonePermissions de diferentes elementos primarios. Sin embargo, puesto que el primer orden de clasificación se ha establecido como descendiente, los elementos secundarios de PhonePermissions de ambos elementos primarios (RowId 861 y RowId 862) se ordenan de esta manera.

```
{parameters:
  {coFilter: {
    object: {
      name:"Person", key: { rowid: 1242 }, order: "lastName",
      object:[
        {name:"Names", order:"-name"},
        {name:"Phone", key: { rowid:861 }, order:"+phoneCountryCd, -phoneNum", object:
        [{name:"PhonePermissions", order:"-column1"}]},
        {name:"Phone", key: {rowid:862}, order:"phoneNum, -phoneCountryCd", object:
        [{name:"PhonePermissions", order:"column1"}]}
      ]}
    }}}}
```

Ejemplo de solicitudes de API

La siguiente solicitud de ejemplo devuelve los detalles del registro raíz en la entidad de negocio de persona.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102
```

La siguiente solicitud de ejemplo devuelve los detalles de un registro secundario con rowId 2. El objeto base secundario es genderCd y el registro secundario se encuentra en un nivel 2.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/genderCd/2?depth=2
```

La siguiente solicitud de ejemplo utiliza el nombre del sistema y la clave de fuente para devolver los detalles del registro raíz:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/SFA:9000000000
```

La siguiente solicitud de ejemplo devuelve los detalles del registro raíz y sus registros XREF:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102?contentMetadata=XREF
```

La siguiente solicitud de ejemplo devuelve los detalles del registro raíz con los nombres ordenados de manera descendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/Names?order=-name
```

La siguiente solicitud de ejemplo devuelve las direcciones de facturación con los valores de RowId ordenados de manera ascendente y las fechas de inicio efectivas ordenadas de manera descendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1242/BillAddresses?
order=rowidObject,-effStartDate
```

Ejemplo de respuesta de API

El siguiente ejemplo muestra los detalles del registro raíz en la entidad de negocio de persona.

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102?
depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "102",
  "label": "DARWENT,JIMMY",
```



```

    "partyType": "Person",
    "statusCd": "A ",
    "lastName": "DARWENT",
    "middleName": "N",
    "firstName": "JIMMY",
    "displayName": "JIMMY N DARWENT",
    "genderCd": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
          "rel": "parent"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd",
          "rel": "self"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd?depth=2",
          "rel": "children"
        }
      ],
      "genderCode": "M"
    },
    "generationSuffixCd": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
          "rel": "parent"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
generationSuffixCd?depth=2",
          "rel": "children"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
generationSuffixCd",
          "rel": "self"
        }
      ],
      "generationSuffixCode": "I"
    }
  }
}

```

El siguiente ejemplo muestra los detalles de un registro secundario del objeto base genderCd.

```

{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102",
      "rel": "parent"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd/2?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/102/
genderCd/2",
      "rel": "self"
    }
  ],
  "rowidObject": "2",
  "label": "LU Gender",
  "genderDisp": "MALE",
  "genderCode": "M"
}

```

Crear registros

La API de REST de creación de registros crea un registro en la entidad de negocio especificada. Envíe los datos del registro en el cuerpo de la solicitud. Utilice la API para la promoción para promover y añadir el registro en la entidad de negocio.

La API utiliza el método POST para crear un registro.

URL de solicitud

La dirección URL de creación de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the source system>
```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP POST a la dirección URL de creación de registros:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the source system>
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the source system>
Content-Type: application/<json/xml>
```

Parámetros de URL

El nombre del sistema de origen es un parámetro obligatorio en la URL de solicitud.

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
systemName	Nombre del sistema de origen.
interactionId	ID de la interacción. Puede agrupar varias solicitudes en una única interacción. Todos los cambios se realizan con el ID de interacción.
startDate y endDate	Especifica el periodo de tiempo en el que el registro es efectivo. Proporcione estos parámetros para un objeto base que tenga habilitada la línea temporal.
validateOnly	Indica si el servicio de entidad de negocio de escritura valida los datos entrantes. El valor predeterminado es false.
recordState	El estado del registro. Utilice el parámetro para especificar el estado inicial del registro. Utilice ACTIVE o PENDING. El valor predeterminado es ACTIVE.
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Cuerpo de la solicitud

Envíe los datos para el registro en el cuerpo de la solicitud de REST. Utilice el formato JSON o el formato XML para enviar los datos. Utilice la API de obtención de metadatos para obtener la estructura de la entidad de negocio y proporcionar los valores de parámetro que se necesitan en el cuerpo de la solicitud.

Cuerpo y encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (interactionId) y el ID de proceso (processId) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```
BES-interactionId: 72200000242000
BES-processId: 15948
```

El cuerpo de la respuesta contiene el registro con los ID de fila generados.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea un registro en la entidad de negocio de persona: La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```
POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person?
systemName=Admin&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9
Content-Type: application/json

{
  firstName: "John",
  lastName: "Smith",
  Phone: {
    item: [
      {
        phoneNumber: "111-11-11"
      }
    ]
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de crear correctamente un registro:

```
BES-interactionId: 72200000242000
BES-processId: 15948
Content-Type: application/json
{
  "Person": {
    "key": {
      "rowid": "2198246",
      "sourceKey": "72200000241000"
    },
    "rowidObject": "2198246",
```

```

    "Phone": {
      "item": [
        {
          "key": {
            "rowid": "260961",
            "sourceKey": "72200000243000"
          },
          "rowidObject": "260961"
        }
      ]
    }
  }
}

```

Actualizar registros

La API de REST de actualización de registros actualiza el registro raíz especificado y sus registros secundarios. Envíe el ID del registro en la URL de solicitud. Envíe el resumen de los cambios en el cuerpo de la solicitud.

Después del cambio, si el registro está en un estado pendiente, use la API de promoción para promover los cambios. Por ejemplo, si la actualización desencadena un flujo de trabajo de revisión, el registro está en el estado pendiente hasta que se completa la revisión.

La API usa el método POST.

Nota: Además, puede usar una versión de PUT simplificada, donde el cuerpo de la solicitud contiene el campo cambiado, pero no un resumen del cambio.

URL de solicitud

La dirección URL de actualización de registros tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?systemName=<name of the source system>

```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP PUT a la dirección URL de actualización de registros:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
systemName=<name of the source system>

```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```

PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
systemName=<name of the source system>
Content-Type: application/json+xml

```

Parámetros de consulta

El nombre del sistema de origen es un parámetro de consulta obligatorio.

Puede usar los siguientes parámetros de consulta en la solicitud:

Parámetro	Descripción
systemName	Nombre del sistema de origen.
interactionId	ID de la interacción. Puede agrupar varias solicitudes en una única interacción. Todos los cambios se realizan con el ID de interacción.

Parámetro	Descripción
startdate y enddate	Especifica el periodo de tiempo en el que el registro es efectivo. Proporcione estos parámetros para un objeto base que tenga habilitada la línea temporal.
validateOnly	Indica si el servicio de entidad de negocio de escritura valida los datos entrantes. El valor predeterminado es false.
recordState	Establece el estado del registro. Utilice ACTIVE, PENDING o DELETED. Por ejemplo, si establece recordState=ACTIVE y la solicitud se ejecuta en un registro eliminado temporalmente, la solicitud devuelve el registro al estado activo.
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Cuerpo de la solicitud

Envíe los datos que se van a actualizar en el cuerpo de la solicitud REST. Utilice el formato JSON o el formato XML para enviar los datos.

Proporcione los nuevos valores de parámetro. Utilice el parámetro \$original para indicar el valor anterior de un parámetro que desee actualizar.

Además, puede usar las siguientes propiedades con los registros secundarios:

Propiedades/ elementos	Tipo	Descripción
MATCH	object	Si desea añadir o quitar candidatos de coincidencia de la tabla de coincidencia para el registro secundario, añada el objeto MATCH al registro secundario.
MERGE	object	Si desea fusionar registros secundarios o quitar candidatos de la fusión, añada el objeto MERGE al registro secundario.

En el siguiente ejemplo de código JSON se cambia el nombre en el registro raíz a Bob:

```
{
  rowidObject: "123",
  firstName: "Bob",
  lastName: "Smith",
  $original: {
    firstName: "John"
  }
}
```

En el siguiente ejemplo de código JSON se quita un candidato de coincidencia de un registro secundario de dirección y se define la fusión para dos registros secundarios de número de teléfono:

```
{
  rowidObject: "123",
  firstName: "Bob",
  lastName: "Smith",
  $original: {
```

```

        firstName: "John"
      }
    }
    Address: { // remove A3 from the matches for A2 in the Address_MTCH table
      item: [
        {
          rowidObject: "A2",
          MATCH: {
            item: [ // to remove matched child records for A2, specify null
              null
            ],
            $original: {
              item: [{key: {rowid: 'A3'}}]
            }
          }
        }
      ]
    }
  }
  Telephone: { // override the matches for the telephone child records
    item:[
      {
        rowid: "T1",
        MERGE: {
          item: [ // to remove merge candidates for T1, specify null
            null,
            null
          ],
          $original: {
            item: [
              {rowid: "T2"},
              {rowid: "T3"}
            ]
          }
        }
      },
      {
        rowid: "T4",
        MERGE: {
          item: [ // to add or override matches, specify matched records
            {rowid: "T2"}
          ],
          $original: {
            item: [
              null
            ]
          }
        }
      }
    ]
  }
}

```

Encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (`interactionId`) y el ID de proceso (`processId`) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```

BES-interactionId: 72200000242000
BES-processId: 15948

```

El cuerpo de la respuesta contiene el registro con los ID de fila generados.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza un registro raíz y su registro secundario en una entidad de negocio. La persona es la entidad de negocio y el teléfono es un objeto base secundario. La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```
PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
systemName=Admin&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9

{
  rowidObject: "233",
  firstName: "BOB",
  lastName: "LLOYD",
  Phone: {
    item: [
      {
        rowidObject: "164",
        phoneNumber: "777-77-77",
        $original: {
          phoneNumber: "(336)366-4936"
        }
      }
    ]
  },
  $original: {
    firstName: "DUNN"
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de actualizar correctamente un registro:

```
BES-interactionId: 72300000001000
BES-processId: 16302
{
  Person: {
    key: {
      rowid: "233",
      sourceKey: "SYS:233"
    },
    rowidObject: "233",
    preferredPhone: {
      key: {}
    }
  }
}
```

Eliminar registros

La API de REST de eliminación de registros elimina un registro raíz en una entidad de negocio. Utilice la API para eliminar los registros secundarios de un registro raíz.

La API utiliza el método DELETE para eliminar un registro.

URL de solicitud

La dirección URL de eliminación de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the root
record>?systemName=Admin
```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP DELETE a la dirección URL de eliminación de registros:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the record>?systemName=Admin
```

Utilice el siguiente formato de URL para eliminar un registro secundario de un registro raíz:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowID of the record>/<child base object>/<rowID of the child record>?systemName=Admin
```

Parámetro de consulta

El nombre del sistema de origen es un parámetro de URL obligatorio. Utilice el parámetro `systemName` para especificar el sistema de origen.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina un registro raíz de la entidad de negocio de persona:

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/292258?systemName=Admin
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la respuesta después de eliminar correctamente un registro raíz de la entidad de negocio de persona:

```
{
  "Person": {
    "key": {
      "rowid": "292258",
      "sourceKey": "WRK50000_7016"
    },
    "rowidObject": "292258"
  }
}
```

Lista Registro

La API de REST de enumeración de registros devuelve la lista de valores de búsqueda o valores de clave externa. Las búsquedas proporcionan datos de referencia y ofrecen una lista de posibles valores para una determinada columna.

La API utiliza el método GET.

Utilice la API para obtener los valores y las descripciones de los códigos de búsqueda. Puede especificar el orden de clasificación de las búsquedas. Utilice el método POST cuando necesite parámetros complejos y en mayor cantidad.

URL de solicitud

La dirección URL de REST de enumeración de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list
```

Utilice el siguiente formato de URL para enumerar los códigos de los valores de búsqueda:

```
http://<host>:<port>/<context>/<database ID>/<lookup table>?action=list&idlabel=<lookup code>%3A<lookup display name>
```


Nota: Utilice la API de obtención de metadatos para obtener la URL exacta y enumerar los valores de búsqueda.

Parámetros de consulta

Para filtrar los resultados, puede anexar los parámetros de consulta a la URL de solicitud.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
suppressLinks	Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false. Por ejemplo, la consulta <code>Person/1242?depth=10&suppressLinks=true</code> mostrará los detalles del registro de hasta 10 niveles secundarios, sin incluir vínculos entre elementos primarios y secundarios en la respuesta.
order	Se utiliza para enumerar los valores de búsqueda en orden ascendente o descendente. Utilice el carácter + como prefijo para especificar el orden ascendente y el carácter - como prefijo para especificar el orden descendente. Si no especifica el prefijo, el conjunto de resultados se ordena en orden ascendente de manera predeterminada. Por ejemplo, la consulta <code>LUNamePrefix?action=list&order=-namePrefixDisp</code> enumera los prefijos por nombre, los cuales se ordenan por los nombres para mostrar de los prefijos en orden descendente.

Solicitud POST para especificar el orden de clasificación

Utilice la solicitud POST para especificar el orden de clasificación de los valores de búsqueda. Incluya los parámetros o los campos en el cuerpo de la solicitud POST.

El siguiente ejemplo describe la manera en la que se debe utilizar la solicitud POST para una operación de lista:

```
http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/ListCO.json
{
  parameters:
  {
    coFilter: {
      object: {
        name: "LUCountry",
        order: "-countryNameDisp"
      }
    }
  }
}
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo enumera los valores de búsqueda:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUGender?action=list
```

La siguiente solicitud de ejemplo enumera los códigos correspondientes al género de los valores de búsqueda:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUGender?
action=list&idlabel=genderCode%3AgenderDisp
```

La siguiente solicitud de ejemplo enumera los prefijos para nombres con sus nombres para mostrar en orden descendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/LUNamePrefix?action=list&order=-namePrefixDisp
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de valores de búsqueda:

```
{
  "firstRecord": 1,
  "pageSize": 2147483647,
  "searchToken": "SVR1.AU5LC",
  "item": [
    {
      "rowidObject": "1",
      "genderDisp": "UNKNOWN",
      "genderCode": "N"
    },
    {
      "rowidObject": "2",
      "genderDisp": "MALE",
      "genderCode": "M"
    },
    {
      "rowidObject": "3",
      "genderDisp": "FEMALE",
      "genderCode": "F"
    }
  ]
}
```

La siguiente respuesta de ejemplo muestra el código para los valores de búsqueda:

```
{
  "item": [
    {
      "id": "F",
      "label": "FEMALE"
    },
    {
      "id": "M",
      "label": "MALE"
    },
    {
      "id": "N",
      "label": "UNKNOWN"
    }
  ],
  "firstRecord": 1,
  "recordCount": 0,
  "pageSize": 2147483647,
  "searchToken": "SVR1.AU5LD"
}
```

Búsqueda Registro

La API de REST de búsqueda de registros busca los valores indexados en un registro raíz que admite búsquedas y en todos los registros secundarios. Puede utilizar filtros y facetas para ver un subconjunto de los resultados de la búsqueda. Las facetas agrupan los resultados de la búsqueda en categorías, mientras que los filtros los acotan. La API busca en todos los campos que se hayan configurado para realizar búsquedas y devuelve los registros que coincidan con los criterios de búsqueda.

La API utiliza el método GET para buscar los índices de los campos en los que se pueden realizar búsquedas.

URL de solicitud

La dirección URL de búsqueda de registro tiene los siguientes formatos:

URL para una búsqueda simple

Utilice la URL siguiente para una búsqueda simple:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de eliminación de registro:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>
```

URL para una búsqueda por campos

Utilice la URL siguiente para una búsqueda por campos:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?fq=<business entity field name>='<business entity field value>'
```

Si especifica un valor numérico negativo para un campo de entidad de negocio, como -120, se podría ver afectada la clasificación de los registros que se devuelve.

URL para una búsqueda de facetas

Utilice la URL siguiente para una búsqueda simple con facetas:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>&facets=<field name>
```

Utilice la URL siguiente para una búsqueda por campos con facetas:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?fq=<business entity field name>='<business entity field value>'&facets=<field name>
```

URL para una búsqueda por filtros

Utilice la URL siguiente para una búsqueda simple con filtros:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<string value>&filters=<field name1>=<field value1> AND <field name2>=<field value2> ...
```

Utilice el parámetro q o fq en la búsqueda.

Codificación de URL

Debe utilizar la codificación URL porque la dirección URL incluye caracteres, como espacios y comillas simples.

El ejemplo siguiente muestra la representación codificada de URL para la dirección URL de búsqueda de registro:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?q=<field name>%3D%27<value of the field>
```

Parámetros de consulta

Utilice los parámetros de consulta `q` o `fq` para proporcionar el valor de cadena para la búsqueda. Los parámetros de consulta `q` y `fq` se excluyen mutuamente. Utilice el parámetro `fq` para una búsqueda por campos. Utilice el operador lógico AND para varias condiciones.

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
<code>q</code>	<p>Especifica el valor de cadena o el término de la búsqueda. La consulta busca ocurrencias del término de la búsqueda en cualquier parte de un registro. Se utiliza en una búsqueda simple.</p> <p>Por ejemplo, la consulta <code>Person?q=STEVE</code> busca registros que incluyan el término STEVE.</p> <p>Para buscar dos o más términos a la vez, incluya los términos entre comillas dobles. Anteponga el carácter <code>+</code> a cada término si desea que los resultados de búsqueda contengan este término. Si el valor de campo contiene un espacio, incluya el valor de campo entre comillas simples.</p> <p>Utilice la siguiente consulta para encontrar una coincidencia exacta con WILLIAM JOHN LAWSON:</p> <pre>Person?q="WILLIAM JOHN LAWSON"</pre> <p>Utilice la siguiente consulta para encontrar WILLIAM, JOHN o LAWSON:</p> <pre>Person?q=WILLIAM JOHN LAWSON</pre> <p>Utilice la siguiente consulta para encontrar WILLIAM, JOHN y LAWSON:</p> <pre>Person?q=WILLIAM JOHN LAWSON&queryOperator=AND</pre>
<code>fq</code>	<p>Especifica el valor de cadena o el término de búsqueda en un campo determinado. La consulta busca el término solo en esa parte de un registro. Se utiliza en una búsqueda dirigida en campos indexados.</p> <p>Por ejemplo, la consulta <code>Person?fq=displayname=STEVE</code> busca registros que incluyan el nombre para mostrar STEVE.</p>
<code>facetas</code>	<p>Especifica los campos que se deberían tratar como facetas o categorías a partir de los cuales se agrupan los resultados de la búsqueda. Especifique solo campos que admitan búsquedas. Se utiliza con los parámetros <code>q</code> y <code>fq</code>. La sintaxis es <code>&facets=FieldName1,FieldName2,FieldNameN</code></p> <p>Por ejemplo, la consulta <code>Person?q=STEVE&facets=department</code> busca personas que incluyan el nombre para mostrar STEVE y agrupa los resultados de la búsqueda por departamentos. La búsqueda muestra los registros de las personas con el nombre para mostrar STEVE y estos registros se agrupan por departamentos.</p>
<code>filtros</code>	<p>Especifica los campos mediante los que puede acotar los resultados de la búsqueda. Especifique solo campos que admitan filtros. Se utiliza con los parámetros <code>q</code> y <code>fq</code>.</p> <p>Por ejemplo, la consulta <code>Person?fq=STEVE&filters=birthdate='1980-11-27T08:00:00Z'</code> busca personas que incluyan el nombre para mostrar STEVE y filtra los resultados de la búsqueda por fecha de nacimiento. La búsqueda muestra los registros de las personas con el nombre para mostrar STEVE y cuya fecha de nacimiento sea el 27 de noviembre de 1980.</p> <p>Nota: Especifique una fecha entre comillas simples.</p>

Parámetro	Descripción
depth	<p>Especifica el número de niveles secundarios que se van a devolver. Especifique 2 para devolver el nodo raíz y sus elementos secundarios directos, y 3 para devolver el nodo raíz, el elemento secundario directo y los elementos secundarios de segundo nivel. Especifique 1 para devolver el nodo raíz únicamente. De manera predeterminada, no se especifica ninguna profundidad.</p> <p>Si no se especifica ninguna profundidad, los resultados de búsqueda devuelven el nodo raíz y los elementos secundarios en los que se ha encontrado una coincidencia para el término de búsqueda.</p> <p>Por ejemplo, la consulta <code>Person?q=STEVE&depth=2</code> busca los resultados que incluyan el término STEVE y devuelve la información sobre el registro raíz y sus elementos secundarios directos.</p>
queryOperator	<p>Especifica si la búsqueda encuentra algunas de las cadenas del término de búsqueda o todas las cadenas del término de búsqueda.</p> <p>El parámetro toma uno de los siguientes valores:</p> <ul style="list-style-type: none"> - OR. Busca alguna de las cadenas que aparecen en el parámetro <code>f o fq</code>. - AND. Busca todas las cadenas que aparecen en el parámetro <code>f o fq</code>. <p>Si no especifica este parámetro, el valor predeterminado es OR.</p> <p>Por ejemplo, la consulta <code>Person?q=WILLIAM JOHN LAWSON&queryOperator=AND</code> busca los registros que contengan WILLIAM, JOHN y LAWSON.</p>
suppressLinks	<p>Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. El valor predeterminado es false.</p> <p>Por ejemplo, la consulta <code>Person?q=STEVE&suppressLinks=true</code> busca registros con el término STEVE y devuelve una respuesta en la que no se incluyen vínculos entre elementos primarios y secundarios.</p>
readSystemFields	<p>Indica si desea devolver los campos del sistema en el resultado. El valor predeterminado es false.</p>
order	<p>Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +.</p> <p>Si desea usar un campo secundario para ordenar los resultados, use el nombre completo del campo. Por ejemplo, <code>BillAddresses.Address.cityName</code>.</p> <p>Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo. Por ejemplo, la consulta <code>Person?order=displayName,-BillAddresses.Address.cityName</code> ordena los resultados por el nombre para mostrar en orden ascendente y, a continuación, por el nombre de ciudad en orden descendente.</p>
maxRecordsToSort	<p>Número máximo de resultados de búsqueda que desea ordenar. El valor predeterminado es 1000.</p>

Especifique un rango con el parámetro `filters`:

Puede usar el parámetro `filters` para acotar las resultados de la búsqueda dentro de un rango determinado. Puede especificar el rango para los campos que admiten filtros de tipos de datos numéricos y de fecha.

Utilice el siguiente formato para el tipo de datos de enteros:

```
fieldName1=[fromValue,toValue]
```

El rango oscila de fromValue a toValue. Asegúrese de que fromValue sea inferior a toValue. Por ejemplo, la consulta `filters=age=[35,45]` acota los resultados de la búsqueda y busca registros en el grupo de edades comprendido entre 35 y 45.

Utilice el siguiente formato para el tipo de datos de fecha:

```
fieldName1=[fromDate,toDate]
```

El rango oscila de fromDate a toDate. Por ejemplo, la consulta `filters=birthdate=[2000-06-12T12:30:00Z,2015-06-12T12:30:00Z]` especifica la fecha de nacimiento entre el 12 de junio de 2000 y el 12 de junio de 2015.

Nota: Cuando especifique un filtro de datos de coincidencia exacta, inclúyalo entre comillas simples. Cuando especifique un rango de fechas, no utilice las comillas.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

Solicitud con el parámetro q

La siguiente solicitud de ejemplo busca la entidad de negocio de persona para los registros con el nombre STEVE.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?q=STEVE
```

Solicitud con el parámetro fq

La siguiente solicitud de ejemplo busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE. El campo `displayName` es un campo indexado.

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE
```

Solicitud con la opción de ordenación

En la siguiente solicitud de ejemplo se busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y los resultados se ordenan por ciudad en orden ascendente:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&order=BillAddresses.Address.cityName
```

Solicitud con el parámetro fq y el operador lógico AND

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y el ID fiscal DM106:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE AND taxId=DM106
```

Solicitud con una faceta

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y acota los resultados agrupándolos en departamentos:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&facets=department
```

Solicitud con un filtro (filtro exacto)

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y filtra por la ciudad y el país especificados:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?fq=displayName=STEVE&filters=cityName=Canberra AND country=Australia
```

Solicitud con un rango de filtros

La siguiente solicitud de muestra busca la entidad de negocio de persona para los registros con el nombre para mostrar STEVE y filtra por el grupo de edades comprendidas entre 35 y 45:

```
GET http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/Person?
fq=displayName=STEVE&filters=age=[35,45] AND cityName=Canberra
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el resultado de una búsqueda del nombre STEVE:

```
{
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 10,
  "item": [
    {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
Person/1443",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
Person/1443?depth=2",
            "rel": "children"
          }
        ],
        "rowidObject": "1443",
        "label": "CRAIG, STEVE",
        "partyType": "Person",
        "lastName": "CRAIG",
        "firstName": "STEVE",
        "taxID": "stevecraig",
        "displayName": "STEVE CRAIG"
      }
    },
    {
      "Person": {
        "link": [
          {
            "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
Person/285",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/
Person/285?depth=2",
            "rel": "children"
          }
        ],
        "rowidObject": "285",
        "label": "PEARSON, STEVE",
        "partyType": "Person",
        "lastName": "PEARSON",
        "firstName": "STEVE",
        "displayName": "STEVE PEARSON"
      }
    }
  ]
}
```

Proveedor de sugerencias

La API de REST del Proveedor de sugerencias devuelve una lista de los términos relacionados para una cadena de búsqueda basada en los datos presentes en su base de datos. Utilice la API para aceptar los caracteres que escribe en un campo de interfaz de usuario y devolver sugerencias para autocompletar lo que está escribiendo. Puede buscar y seleccionar la cadena desde la lista de sugerencias. Utilice la API del Proveedor de sugerencias para los campos que admiten búsquedas.

La API utiliza el método GET.

Nota: Para usar la API para que proporcione una lista de sugerencias para autocompletar en un campo que admite búsquedas, establezca la propiedad `suggester` del campo como verdadera y vuelva a indexar los datos.

URL de solicitud

La URL de REST del Proveedor de sugerencias tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?suggest=<string>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>?suggest=<string>
```

Nota: Si utiliza el motor de búsqueda Solr, el índice del proveedor de sugerencias tendrá que reconstruirse cada vez que se reinicie el servidor de procesos. Para garantizar que el índice se reconstruya cada vez que se reinicie un servidor de procesos, establezca el parámetro `buildIndex` en `true`:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?  
suggest=<string>&buildIndex=true
```

Parámetros de consulta

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
<code>suggest</code>	Obligatorio. Especifica la cadena para la que desea recibir sugerencias.
<code>recordsToReturn</code>	Especifica el número de filas que se van a devolver.
<code>buildIndex</code>	Opcional. Indica si se va a crear el índice. Cuando reinicie el sistema, ajuste este parámetro como verdadero para crear explícitamente el índice de la colección. Este parámetro puede quedar en desuso en versiones futuras.

Ejemplo de solicitud de API

El siguiente ejemplo devuelve una lista de las sugerencias que puede usar en una interfaz de usuario:

```
GET http://localhost:8080/cmx/cs/localhost-info-DS_UI1/Person.json?suggest=Abhinav
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de sugerencias:

```
{  
  term: [2]  
  "abhinav goel"  
  "abhinav gupta"
```



```
}
```

Obtener metadatos de BPM

La API de REST para la obtención de metadatos de BPM devuelve los tipos de tarea y dos indicadores que especifican si la herramienta del flujo de trabajo de BPM puede realizar el servicio para obtener el linaje de la tarea y los servicios de administración.

La API utiliza el método GET.

URL de solicitud

La URL para la obtención de metadatos de BPM tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/BPMMetadata
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de metadatos de BPM:

```
GET http://<host>:<port>/<context>/<database ID>/BPMMetadata
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve información sobre los tipos de tarea y la herramienta de flujo de trabajo de BPM:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/BPMMetadata
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los tipos de tarea y el valor de los dos indicadores para la herramienta del flujo de trabajo de BPM:

```
{
  "parameters": {
    "doesSupportAdministration": true,
    "doesSupportLineage": true,
    "doesSupportAttachments": true,
    "maximumAttachmentFileSizeInMb": 20,
    "taskTypes": {
      "taskTypes": [
        {
          "name": "Merge",
          "label": "Merge"
        },
        {
          "name": "FinalReview",
          "label": "FinalReview"
        },
        {
          "name": "Update",
          "label": "Update"
        },
        {
          "name": "Notification",
          "label": "Notification"
        },
        {
          "name": "ReviewNoApprove",
          "label": "ReviewNoApprove"
        },
        {
          "name": "Unmerge",
          "label": "Unmerge"
        }
      ]
    }
  }
}
```

```

    }
  }
]

```

Lista de tareas

La API para la obtención de una lista de tareas devuelve una lista de tareas del flujo de trabajo. Un flujo de trabajo define las actividades en un proceso de negocio y las rutas de ejecución a través de las actividades. Cada actividad se denomina una tarea.

La API utiliza el método GET para devolver una lista de tareas clasificada y paginada.

URL de solicitud

La URL para la obtención de una lista de tareas tiene el siguiente formato.

```
http://<host>:<port>/<context>/<database ID>/task
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de una lista de tareas:

```
GET http://<host>:<port>/<context>/<database ID>/task
```

Puede agregar encabezados HTTP a la solicitud.

Parámetros de consulta

Utilice los campos de datos de tarea como parámetros de consulta para filtrar la lista de tareas.

Puede utilizar los siguientes parámetros de consulta:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte la <i>Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo.
creator	Usuario que crea la tarea.

Parámetro	Descripción
createDateBefore y createDateAfter	Intervalo de fechas. Las tareas se pueden filtrar por el campo createDate.
dueDateBefore y dueDateAfter	Intervalo de fechas. Las tareas se pueden filtrar por el campo dueDate.

Utilice los parámetros de consulta como los pares name-value en la URL de la solicitud.

En el siguiente ejemplo se muestra cómo se pueden utilizar los parámetros de consulta para filtrar tareas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task?
recordsToReturn=100&owner=sergey&status=OPEN
```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Parámetros de clasificación

Para determinar el orden de clasificación en la respuesta de la API de REST, utilice el parámetro de clasificación genérico y proporcione una lista de campos de tarea separados por comas. Puede especificar el orden de clasificación para cada campo. Utilice el signo de guión (-) para especificar el orden descendiente. El orden de clasificación predeterminado es el ascendente.

El siguiente ejemplo muestra cómo se clasifican los resultados:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task?
recordsToReturn=100&owner=sergey&status=OPEN&sort=-priority
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera la lista de tareas:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task
```

La solicitud no contiene un cuerpo.

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de tareas en formato JSON:

```
{
  "firstRecord": 1,
  "recordCount": 10,
  "pageSize": 10,
  "task": [
    {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:15443",
          "rel": "self"
        }
      ],
      "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "taskAction": [
          {

```

```

        "name": "Escalate",
        "label": "Escalate",
        "nextTaskType": "AVOSBeFinalReview",
        "comment": "AS_REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
    },
    {
        "name": "Reject",
        "label": "Reject",
        "nextTaskType": "AVOSBeUpdate",
        "comment": "MANDATORY",
        "attachment": "MANDATORY",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
    },
    {
        "name": "Disclaim",
        "label": "Disclaim",
        "nextTaskType": "AVOSBeReviewNoApprove",
        "comment": "AS_REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
    }
    ],
    "pendingBVT": true,
    "updateType": "PENDING"
},
"taskId": "urn:b4p2:15443",
"title": "Review changes in SMITH,SMITH",
"dueDate": "2015-07-15T21:45:59-07:00",
"status": "OPEN",
"priority": "NORMAL",
"businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15440",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:15440",
    "title": "Review changes in SMITH,JOHN",
    "dueDate": "2015-07-15T21:37:50-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15437",
            "rel": "self"
        }
    ],
    "taskType": {

```

```

        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "taskAction": [
            {
                "name": "Reject",
                "label": "Reject",
                "nextTaskType": "AVOSBeUpdate",
                "comment": "AS_REQUIRED",
                "attachment": "MANDATORY",
                "manualReassign": false,
                "closeTaskView": true,
                "cancelTask": false
            }
        ],
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:15437",
    "title": "Review changes in SMITH,JOHN",
    "dueDate": "2015-07-15T21:34:32-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14820",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:14820",
    "title": "Review changes in STAS,STAS",
    "dueDate": "2015-07-14T10:40:51-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14809",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:14809",
    "title": "Review changes in ,93C8ORScoFSA687",
    "dueDate": "2015-07-14T08:28:15-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [

```

```

        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14609",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:14609",
    "title": "Review changes in A8,A8",
    "dueDate": "2015-07-13T08:40:11-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14425",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:14425",
    "title": "Review changes in A7,A7",
    "dueDate": "2015-07-10T14:11:02-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14422",
            "rel": "self"
        }
    ],
    "taskType": {
        "name": "ReviewNoApprove",
        "label": "Review No approve",
        "taskKind": "REVIEW",
        "pendingBVT": true,
        "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:14422",
    "title": "Review changes in A6,A6",
    "dueDate": "2015-07-10T13:54:09-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
},
{
    "link": [
        {
            "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/
urn:b4p2:14415",
            "rel": "self"
        }
    ]
}

```

```

    ],
    "taskType": {
      "name": "ReviewNoApprove",
      "label": "Review No approve",
      "taskKind": "REVIEW",
      "pendingBVT": true,
      "updateType": "PENDING"
    },
    "taskId": "urn:b4p2:14415",
    "title": "Review changes in A5,A5",
    "dueDate": "2015-07-10T13:51:12-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
  },
  {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:14355",
        "rel": "self"
      }
    ],
    "taskType": {
      "name": "Notification",
      "label": "Notification",
      "taskKind": "REVIEW",
      "pendingBVT": false,
      "updateType": "ACTIVE"
    },
    "taskId": "urn:b4p2:14355",
    "title": "Review changes in A4,A4",
    "dueDate": "2015-07-10T10:31:57-07:00",
    "status": "OPEN",
    "priority": "NORMAL",
    "businessEntity": "Person"
  }
]
}

```

Leer tarea

La API de REST para la lectura de tareas devuelve los detalles de una tarea, como son el tipo de tarea, su prioridad y estado.

La API utiliza el método GET.

URL de solicitud

La dirección URL para la lectura de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Nota: Utilice la API para la obtención de una lista de tareas para obtener el ID de la tarea.

Haga la siguiente solicitud HTTP GET a la URL para la lectura de tareas:

```
GET http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los detalles de una tarea:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:16605
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los detalles de la tarea:

```
{
  "taskType": {
    "name": "ReviewNoApprove",
    "label": "Review No Approve",
    "taskKind": "REVIEW",
    "taskAction": [
      {
        "name": "Escalate",
        "label": "Escalate",
        "nextTaskType": "AVOSBeFinalReview",
        "comment": "AS REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      },
      {
        "name": "Reject",
        "label": "Reject",
        "nextTaskType": "AVOSBeUpdate",
        "comment": "MANDATORY",
        "attachment": "MANDATORY",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      },
      {
        "name": "Disclaim",
        "label": "Disclaim",
        "nextTaskType": "AVOSBeReviewNoApprove",
        "comment": "AS REQUIRED",
        "attachment": "NEVER",
        "manualReassign": false,
        "closeTaskView": true,
        "cancelTask": false
      }
    ],
    "pendingBVT": true,
    "updateType": "PENDING"
  },
  "taskId": "urn:b4p2:16605",
  "processId": "16603",
  "title": "Review changes in HERNANDEZ,ALEJANDRO",
  "dueDate": "2015-07-23T01:18:39.125-07:00",
  "status": "OPEN",
  "priority": "NORMAL",
  "taskRecord": [
    {
      "businessEntity": {
        "key": {
          "rowid": "114",
          "sourceKey": "SYS:114"
        },
        "name": "Person"
      }
    },
    {
      "businessEntity": {
        "key": {
          "rowid": "114",
          "sourceKey": "SYS:114",
          "rowidXref": "4680363"
        },
        "name": "Person.XREF"
      }
    }
  ],
}
```



```

"creator": "avos",
"createDate": "2015-07-16T01:18:46.148-07:00",
"attachments": [
  {
    "id": "urn.b4p2:22203::file1.txt",
    "name": "file1.txt",
    "contentType": "text/plain",
    "creator": "admin",
    "createDate": "2018-02-26T14:31:05.590-05:00"
  }
],
"businessEntity": "Person",
"interactionId": "72340000003000",
"orsId": "localhost-orcl-DS_UI1"
}

```

Crear tarea

La API de REST para la creación de tareas crea una tarea e inicia un flujo de trabajo.

La API utiliza el método POST para crear una tarea y devolver el ID del proceso de flujo de trabajo que contiene la tarea.

URL de solicitud

La dirección URL para la creación de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task
```

Haga la siguiente solicitud POST HTTP a la URL para la creación de tareas:

```
POST http://<host>:<port>/<context>/<database ID>/task
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/task
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Especifique los atributos de tarea cuando cree la tarea. Utilice el formato JSON o el formato XML para enviar los datos de la tarea en la solicitud.

En la siguiente tabla se describen los parámetros de tarea del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte la <i>Guía de implementación de Multidomain MDM Data Director</i> .
owner	El usuario al que el creador asigna la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
adjuntos	Adjuntos de la tarea.

Parámetro	Descripción
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo. El valor predeterminado es Normal.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se crea la tarea.
orsId	El ID del Almacén de referencias operativas (ORS) como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	El ID de tipo de tarea de ActiveVOS®. Para obtener más información, véase <i>Guía de implementación de Multidomain MDM Data Director</i> .
taskRecord	El registro raíz de objeto de negocio o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity	Nombre de la entidad de negocio al que pertenece taskRecord.
interactionId	ID de la interacción. Use el ID de interacción para mantener la relación de contexto de tarea entre una tarea y los registros.
groups	Asigne una tarea a todos los usuarios de los grupos de usuarios especificados. Defina los grupos de usuarios en la consola de MDM Hub. Especifique los grupos como una matriz.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```
taskRecord: [{
  businessEntity: {
    name: "Person",
    key: {
      rowid: "233",
    }
  }
}]
```

El cuerpo de la solicitud presenta el siguiente formato:

```
{
  taskType: {name:"name of the task"},
  owner: "user who performs the task",
  title: "title of the task",
  comments: "description of the task",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "date to complete the task",
  status: "status of the task",
  priority: "priority of the task",
  creator: "use who creates the task",
  createDate: "date on which the task is created",
  updatedBy: "user who last updated the task",
  lastUpdateDate: "date on which the task was last updated",
}
```

```

    businessEntity: "name of the business entity",
    interactionID: "ID of an interaction",
    groups: ["group name A", "group name B", ...],
    orsId: "database ID",
    processId: "ActiveVOS task type ID",
    taskRecord: [{
      businessEntity:{
        name: "name of the business entity",
        key:{
          rowid: "rowId of the record", //Use the rowId or the source system and
          source key to identify the record.
        }
      }
    }]
  }
}

```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea una tarea para un registro raíz:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task
{
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "",
  owner: "manager",
  title: "Smoke test task",
  comments: "Smoke testing",
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: "IDDUpdateWithApprovalTask",
  taskRecord: [{
    businessEntity:{
      name: "Person",
      key:{
        rowid: "123"
      }
    }
  }]
}

```

Ejemplo de respuesta de API

El ejemplo siguiente muestra la respuesta al crear correctamente una tarea. La API devuelve el ID del proceso de flujo de trabajo que contiene la tarea.

```

{
  "parameters": {
    "processId": "15827"
  }
}

```

Actualizar tarea

La API de REST para la actualización de tarea actualiza una única tarea.

La API utiliza el método PATCH para actualizar algunos campos de las tareas y el método PUT para actualizar la tarea completa. Proporcione el ID de la tarea como el parámetro URL.

URL de solicitud

La dirección URL para la actualización de tareas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Nota: Utilice la API para la obtención de una lista de tareas para obtener el ID de la tarea.

Haga la siguiente solicitud HTTP PUT a la URL para la actualización de tareas con el fin de actualizar la tarea completada:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Haga la siguiente solicitud HTTP PATCH a la URL para la actualización de tareas con el fin de actualizar algunos campos de la tarea:

```
PATCH http://<host>:<port>/<context>/<database ID>/task/<taskId>
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>  
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Utilice la API para la lectura de tareas para obtener los detalles de la tarea. Especifique los atributos de tarea cuando actualice la tarea. Utilice el formato JSON o el formato XML para enviar los datos que se van a actualizar en la solicitud.

En la siguiente tabla se describen los parámetros de tarea del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte <i>la Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
adjuntos	Adjuntos de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.

Parámetro	Descripción
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo. El valor predeterminado es Normal.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se creó la tarea.
updatedBy	Usuario que actualiza la tarea.
lastUpdateDate	Fecha de la última actualización de la tarea.
orsId	El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
taskRecord	El registro raíz o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity name	Nombre de la entidad de negocio al que pertenece taskRecord.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```
taskRecord: [{
  businessEntity: {
    name: 'Person',
    key: {
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]
```

En una solicitud de parche (PATCH), el cuerpo de la solicitud contiene los campos de tarea que desea cambiar. Es posible cambiar el título de la tarea, su prioridad, la fecha de vencimiento y el propietario.

En una solicitud de colocación (PUT), el cuerpo de la solicitud debe incluir todos los campos de tarea. Utilice el siguiente cuerpo de solicitud para crear una solicitud PUT:

```
{
  taskType: {name:"name of the task"},
  taskId: "ID of the task",
  owner: "user who performs the task",
  title: "title of the task",
  comments: "description of the task",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "date to complete the task",
  status: "status of the task",
  priority: "priority of the task",
  creator: "use who creates the task",
  createDate: "date on which the task is created",
  updatedBy: "user who last updated the task",
  lastUpdateDate: "date on which the task was last updated",
  businessEntity: "name of the business entity",
  orsId: "database ID",
}
```

```

    processId: 'ActiveVOS task type ID',
    taskRecord: [{
      businessEntity:{
        name: 'name of the business entity',
        key:{
          rowid: 'rowId of the record',//Use the rowId or the source system and source
key to identify the record.
          systemName: '',
          sourceKey: ''
        }
      }
    }]
  }
}

```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

La siguiente solicitud PUT de ejemplo actualiza una tarea completa:

```

PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934
{
  taskType: {name:"UpdateWithApprovalWorkflow"},
  taskId: "urn:b4p2:15934",
  owner: "John",
  title: "Smoke test task - updated",
  comments: "Smoke testing - updated",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "2015-08-15T00:00:00",
  status: "OPEN",
  priority: "HIGH",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: '3719',
  taskRecord: [{
    businessEntity:{
      name: 'Person',
      key:{
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  }]
}

```

La siguiente solicitud PATCH de ejemplo actualiza algunos campos de tarea:

```

PATCH http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934
{
  processId: "3719",
  priority: "HIGH",
  owner: "John"
}

```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al actualizar correctamente una tarea. El cuerpo de la respuesta está vacío.

Tarea completada

La API de REST para tareas completadas cierra un flujo de trabajo de tareas después de que finalicen todas las tareas del flujo de trabajo. Utilice la API para cerrar un flujo de trabajo después de procesar todos los registros relacionados con la tarea. Por ejemplo, cuando elija fusionar candidatos, podrá crear una tarea que inicie el flujo de trabajo de la fusión. La tarea de fusión se completará una vez que haya obtenido una vista previa de cada candidato y lo haya fusionado o lo haya marcado como no coincidente. Utilice la API para cerrar el flujo de trabajo de la fusión.

La API utiliza el método PUT.

URL de solicitud

La URL para la tarea completada tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
```

Haga la siguiente solicitud HTTP PUT a la URL para la tarea completada:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=complete
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Envíe los detalles de la tarea en el cuerpo de la solicitud. Utilice la API para la lectura de tareas para obtener los detalles de la tarea.

En la siguiente tabla se describen los parámetros de tarea del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte <i>la Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: <ul style="list-style-type: none">- Abierto: la tarea no se ha iniciado o está en curso.- Cerrado: la tarea ha finalizado o se ha cancelado.

Parámetro	Descripción
priority	Nivel de importancia de la tarea.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se creó la tarea.
updatedBy	Usuario que actualiza la tarea.
lastUpdateDate	Fecha de la última actualización de la tarea.
orsId	El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
taskRecord	El registro raíz o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity name	Nombre de la entidad de negocio al que pertenece taskRecord.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```

taskRecord: [{
  businessEntity: {
    name: 'Person',
    key: {
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]

```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo completa el flujo de trabajo de fusión:

```

PUT http://localhost:8080/cm/cs/localhost-orcl-DS_UI1/task/urn:b4p2:20210?
action=complete

{
  "taskType": {"name": "Merge"},
  "taskId": "urn:b4p2:20210",
  "owner": "admin",
  "dueDate": "2015-08-14T17:00:00-07:00",
  "status": "OPEN",
  "priority": "NORMAL",
  "creator": "admin",
  "createDate": "2015-06-15T00:00:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2015-06-15T00:00:00",
  "businessEntity": "Person",
  "orsId": "localhost-orcl-DS_UI1",
  "processId": "20208",
  "taskRecord": [{

```



```

        "businessEntity":{
            "name": "Person",
            "key":{
                "rowid": "233",
                "systemName": "",
                "sourceKey": ""
            }
        }
    }
}
]]
}

```

Ejemplo de respuesta de API

La API devuelve una respuesta de 200 OK al completar correctamente un flujo de trabajo de tareas. El cuerpo de la respuesta está vacío.

Ejecutar acción de tarea

La API de REST para la ejecución de acciones de tarea restablece una tarea en el flujo de trabajo para procesarla más adelante. Cada tipo de tarea tiene un conjunto de acciones de tarea y un flujo de trabajo que especifica la secuencia de tareas. Al ejecutar una acción de tarea, la tarea se desplaza al siguiente paso del flujo de trabajo. Si a una acción de tarea no le sigue ninguna tarea después, el flujo de trabajo finalizará cuando ejecute la acción de tarea.

La API utiliza el método POST para realizar acciones como, por ejemplo, aprobar, escalar o cancelar una tarea.

URL de solicitud

La siguiente URL especifica el formato de la URL para la ejecución de una acción de tarea:

```
http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
```

Nota: Utilice la API para la obtención de una lista de tareas para obtener el ID de la tarea.

Haga la siguiente solicitud HTTP POST a la URL para la ejecución de una acción de tarea:

```
POST http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
```

Si desea editar la tarea antes de ejecutar la acción de la tarea, agregue el encabezado Content-Type para especificar el tipo de medio de los datos de la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/task/<taskId>?action=<taskAction>
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Proporcione los datos de la tarea en el cuerpo de la solicitud si desea cambiar los detalles de la tarea antes de ejecutar la acción de esta.

En la tabla siguiente se describen los parámetros del cuerpo de la solicitud:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Utilice el atributo name para especificar el tipo de tarea. Para obtener más información sobre tipos de tarea, consulte <i>la Guía de implementación de Multidomain MDM Data Director</i> .
taskId	ID de la tarea.

Parámetro	Descripción
owner	Usuario que realiza la tarea.
title	Breve descripción de la tarea.
comments	Comentarios de la tarea.
adjuntos	Adjuntos de la tarea.
dueDate	La fecha en la que el propietario debe completar la tarea.
status	Estado de la tarea en el flujo de trabajo. Utilice uno de los siguientes dos valores: - Abierto: la tarea no se ha iniciado o está en curso. - Cerrado: la tarea ha finalizado o se ha cancelado.
priority	Nivel de importancia de la tarea. Utilice uno de los siguientes valores: alto, normal y bajo.
creator	Usuario que crea la tarea.
createDate	La fecha en la que se creó la tarea.
updatedBy	Usuario que actualiza la tarea.
lastUpdateDate	Fecha de la última actualización de la tarea.
businessEntity	Nombre de la entidad de negocio.
orsId	El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.
processId	ID del proceso de flujo de trabajo que contiene la tarea.
taskRecord	El registro raíz o el registro de referencia cruzada asociado a la tarea. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar el registro.
businessEntity name	Nombre de la entidad de negocio al que pertenece taskRecord.

En el siguiente código de ejemplo se utiliza el ID de fila (rowId) para especificar el registro de tarea (taskRecord):

```

taskRecord: [{
  businessEntity: {
    name: 'Person',
    key: {
      rowid: '233',
      systemName: '',
      sourceKey: ''
    }
  }
}]

```

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo cancela una tarea y finaliza el flujo de trabajo:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/task/urn:b4p2:15934?
taskAction=Cancel

{
  taskType: {
    name:"UpdateWithApprovalWorkflow",
    taskAction: [{name: "Cancel"}]
  },
  taskId: "urn:b4p2:15934",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  attachments: [
    {
      id: "TEMP_SVR1.1VDVS"
    }
  ],
  dueDate: "2015-06-15T00:00:00",
  status: "OPEN",
  priority: "NORMAL",
  creator: "admin",
  createDate: "2015-06-15T00:00:00",
  updatedBy: "admin",
  lastUpdateDate: "2015-06-15T00:00:00",
  businessEntity: "Person",
  orsId: "localhost-orcl-DS_UI1",
  processId: '3685',
  taskRecord: [{
    businessEntity:{
      name: 'Person',
      key:{
        rowid: '123',
        systemName: '',
        sourceKey: ''
      }
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al ejecutar correctamente una acción de tarea. El cuerpo de la respuesta está vacío.

Lista de usuarios asignables

La API de REST para la obtención de una lista de usuarios asignables devuelve una lista de los usuarios a los que se les pueden asignar las tareas que pertenecen a un tipo de tarea. Utilice la API para obtener los usuarios de destino de una tarea.

La API utiliza el método GET.

URL de solicitud

La URL para la obtención de una lista de usuarios asignables tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/user?taskType=<task type>&businessEntity=<business entity name>
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de una lista de usuarios asignables:

```
GET http://<host>:<port>/<context>/<database ID>/user?taskType=<task type>&businessEntity=<business entity name>
```

Parámetros de consulta

En la siguiente tabla se enumeran los parámetros obligatorios en la dirección URL:

Parámetro	Descripción
taskType	Conjunto de acciones que puede realizar en un registro. Los tipos de tarea incluyen la actualización con aprobación, la actualización con aprobación opcional, la fusión, la anulación de fusión, la revisión sin aprobación, la revisión final y la actualización del registro rechazado.
businessEntity	Nombre de la entidad de negocio.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera una lista de los usuarios asignables:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/user.json?taskType=ReviewNoApprove&businessEntity=Person
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la lista de usuarios asignables para el tipo de tarea ReviewNoApprove:

```
{
  "users": {
    "user": [
      {
        "userName": "admin"
      }
    ]
  },
  "roles": {}
}
```

Enumerar metadatos de archivo

La API de REST de enumeración de metadatos de archivo devuelve una lista de los metadatos de archivo en un almacenamiento.

Utilice la API de REST de enumeración de metadatos de archivo con un almacenamiento del tipo BPM o TEMP.

La API usa el método GET.

URL de solicitud

La dirección URL de enumeración de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de enumeración de metadatos de archivo:

```
GET http://<host>:<port>/<context>/<database ID>/<storage>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo recupera una lista de los metadatos de archivo en un almacenamiento TEMP:

```
GET http://localhost:8080/cmxf/file/localhost-orcl-MDM_SAMPLE/TEMP
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra una lista de los metadatos de archivo:

```
{
  files: [
    {
      "fileId": "TEMP_SVR1.1VDVS",
      "fileName": "file1.txt",
      "fileType": "text",
      "fileContentType": "text/plain",
    },
    {
      "fileId": "TEMP_SVR1.2ESDS",
      "fileName": "image1.png",
      "fileType": "image",
      "fileContentType": "image/png",
    },
    ...
  ]
}
```

Crear metadatos de archivo

La API de REST de creación de metadatos de archivo crea metadatos para un archivo y devuelve un ID de archivo para el archivo.

El ID de archivo se puede usar para cargar el archivo, adjuntarlo, actualizarlo, descargarlo y eliminarlo.

Utilice la API de REST de creación de metadatos de archivo con un almacenamiento del tipo DB o TEMP.

La API usa el método POST.

URL de solicitud

La dirección URL de creación de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>
```

Haga la siguiente solicitud HTTP POST a la dirección URL de creación de metadatos de archivo:

```
POST http://<host>:<port>/<context>/<database ID>/<storage>
```

Cuerpo de la solicitud

Especifique los metadatos del archivo.

En la tabla siguiente se describen los parámetros de los metadatos de archivo del cuerpo de la solicitud:

Parámetro	Descripción
fileName	Nombre del archivo. Por ejemplo, file.txt.
fileType	Categoría del tipo de archivo. Por ejemplo, text o image.
fileContentType	Tipo de contenido del archivo. El tipo de contenido consta de un tipo y un subtipo separados por /. Por ejemplo, image/png.

Nota: Los parámetros que se requieren para la solicitud de la API de REST de creación de metadatos de archivo son específicos del almacenamiento.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea los metadatos de un archivo en un almacenamiento TEMP:

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP

{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

Ejemplo de respuesta de API

El ejemplo siguiente muestra la respuesta al crear correctamente los metadatos de un archivo en un almacenamiento TEMP. La API devuelve el ID de archivo para el archivo.

```
TEMP_SVR1.1VDVS
```

Nota: El formato del ID de archivo es <storage type>_<uniqueID>.

Obtener metadatos de archivo

La API de REST de obtención de metadatos de archivo devuelve los metadatos de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de obtención de metadatos de archivo con un almacenamiento del tipo BPM, BUNDLE, DB o TEMP.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de metadatos de archivo:

```
GET http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve los metadatos de un archivo con el ID de archivo TEMP_SVR1.1VDVS en el almacenamiento TEMP:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

La siguiente solicitud de ejemplo devuelve los metadatos del archivo de paquete de recursos con el ID de archivo besMetadata en el almacenamiento BUNDLE:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los metadatos de un archivo con el ID de archivo TEMP_SVR1.1VDVS en un almacenamiento TEMP:

```
{
  "fileName": "file1.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

La siguiente respuesta de ejemplo muestra los metadatos del archivo de paquete de recursos con el ID de archivo besMetadata en un almacenamiento BUNDLE:

```
{
  "fileName": "besMetadata.zip",
  "fileType": "BES Metadata Bundle",
  "fileContentType": "application/zip",
  "digest": "a08c5d97da7e6a780ed7c427ff14a8d2d396438cd65b654ad67424e226f64a41"
}
```

Actualizar metadatos de archivo

La API de REST de actualización de metadatos de archivo actualiza los metadatos de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de actualización de metadatos de archivo con un almacenamiento del tipo DB o TEMP.

La API utiliza el método PUT.

URL de solicitud

La dirección URL de actualización de metadatos de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Haga la siguiente solicitud HTTP PUT a la dirección URL de actualización de metadatos de archivo:

```
PUT
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo actualiza los metadatos de un archivo con el ID de archivo TEMP_SVR1.1VDVS en un almacenamiento TEMP:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

```
{
  "fileName": "file2.txt",
  "fileType": "text",
  "fileContentType": "text/plain"
}
```

La siguiente solicitud de ejemplo actualiza los metadatos de un archivo con el ID de archivo `DB_SVR1.OJU1` en un almacenamiento DB:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1
{
  "fileName": "Document_2.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al actualizar correctamente los metadatos de archivo. El cuerpo de la respuesta está vacío.

Cargar contenido de archivo

La API de REST de carga de contenido de archivo carga el contenido de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de carga de contenido de archivo con un almacenamiento del tipo `BUNDLE`, `DB` o `TEMP`.

La API utiliza el método `PUT`.

URL de solicitud

La dirección URL de carga de contenido de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Haga la siguiente solicitud HTTP `PUT` a la dirección URL de carga de contenido de archivo:

```
PUT http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo carga el contenido de un archivo con el ID de archivo `TEMP_SVR1.1VDVS` en un almacenamiento `TEMP`:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS/content
Test attachment content: file 1
```

La siguiente solicitud de ejemplo carga el contenido de un archivo con el ID de archivo `DB_SVR1.OJU1` en un almacenamiento `DB`:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1/content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

La siguiente solicitud de ejemplo carga el contenido de un archivo de paquete de recursos con el ID de archivo `besMetadata` en un almacenamiento `BUNDLE`:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/content
Content-Type: application/octet-stream
Body: binary stream - zip file with besMetadata bundle
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al cargar correctamente el contenido de un archivo. El cuerpo de la respuesta está vacío.

Obtener contenido de archivo

La API de REST de obtención de contenido de archivo devuelve el contenido de un archivo que está asociado a un ID de archivo.

Utilice la API de REST de obtención de contenido de archivo con un almacenamiento del tipo BPM, BUNDLE, DB o TEMP.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de contenido de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de contenido de archivo:

```
GET http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>/content
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve el contenido de un archivo con el ID de archivo

urn:b4p2:22203::file1.txt en un almacenamiento BPM:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BPM/urn:b4p2:22203::file1.txt/content
```

Nota: Utilice la API de REST de lectura de tareas para recuperar el ID de archivo de un adjunto de tarea en un almacenamiento BPM.

La siguiente solicitud de ejemplo devuelve el contenido de un archivo con el ID de archivo DB_SVR1.OJU1 en un almacenamiento DB:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1/content
```

Nota: Utilice la API de REST de lectura de registros para recuperar el ID del archivo que se adjunta a un registro.

La siguiente solicitud de ejemplo devuelve el contenido del archivo de paquete de recursos con el ID de archivo besMetadata en un almacenamiento BUNDLE:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/content
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el contenido de un archivo TXT en un almacenamiento BPM:

```
Test attachment content: file 1
```

La siguiente respuesta de ejemplo muestra el contenido del archivo de paquete de recursos en un almacenamiento BUNDLE:

```
Content-Disposition → attachment; filename=besMetadata.zip  
Content-Type → application/octet-stream  
Transfer-Encoding → chunked
```

Suprimir archivo

La API de REST de eliminación de archivos elimina el archivo que está asociado a un ID de archivo, el cual incluye el contenido y los metadatos del archivo.

Utilice la API de REST de eliminación de archivos con un almacenamiento del tipo BUNDLE, DB o TEMP.

La API usa el método DELETE.

URL de solicitud

La dirección URL de eliminación de archivo tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Haga la siguiente solicitud HTTP DELETE a la dirección URL de eliminación de archivo:

```
DELETE http://<host>:<port>/<context>/<database ID>/<storage>/<fileId>
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina el archivo que está asociado al ID de archivo `TEMP_SVR1.1VDVS` en un almacenamiento `TEMP`, incluidos el contenido y los metadatos de archivo:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS
```

La siguiente solicitud de ejemplo elimina el archivo que está asociado al ID de archivo `DB_SVR1.OJU1` en un almacenamiento `DB`, incluidos el contenido y los metadatos de archivo:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU1
```

La siguiente solicitud de ejemplo elimina el archivo de paquete de recursos con el ID de archivo `besMetadata` en un almacenamiento `BUNDLE`, incluidos el contenido y los metadatos de archivo:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata
```

Ejemplo de respuesta de API

La API devuelve un código de respuesta de 200 OK al eliminar correctamente un archivo. El cuerpo de la respuesta está vacío.

Vista previa de promoción

La API de REST de vista previa de promoción devuelve una vista previa de un registro resultante si promueve los cambios pendientes.

La API utiliza el método GET. Puede ver qué aspecto tendría un registro si aplica los cambios pendientes en el registro. La respuesta de la API contiene el registro con los valores nuevos y un resumen de cambios con los valores anteriores. La API no devolverá información sobre los datos que elimine. Proporcione el ID de interacción de los cambios pendientes en la URL.

URL de solicitud

La URL para la vista previa de la promoción tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID><business entity>/<rowId>?  
action=previewPromote&interactionID=<interaction ID>
```

Haga la siguiente solicitud HTTP GET a la URL para la vista previa de la promoción:

```
GET http://<host>:<port>/<context>/<database ID><business entity>/<rowId>?  
action=previewPromote&interactionID=<interaction ID>
```

Parámetros de consulta

El ID de interacción de los cambios pendientes es un parámetro obligatorio en la URL.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
contentMetadata	Metadatos para la vista previa de fusión. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- BVT. Especifique el valor de rowid del registro que contiene el valor de máxima confianza para su uso en la vista previa de fusión. Devuelve información sobre el registro de referencias cruzadas y el ID de registro original.- MERGE. Especifique los valores de rowid de los registros que se van a fusionar. Devuelve información sobre cómo se han fusionado los registros descendientes.
interactionId	El ID de interacción de los cambios pendientes.
effectiveDate	Opcional. Fecha en la que desea obtener una vista previa de los cambios. Utilice el parámetro para objetos base que tengan habilitada la línea temporal.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea una vista previa de un registro raíz en la entidad de negocio de persona:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
action=previewPromote&interactionId=72300000001000
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo devuelve una vista previa del registro con los valores nuevos y un resumen de cambios de los valores anteriores:

```
{
  "rowidObject": "233",
  "creator": "admin",
  "createDate": "2008-08-12T02:15:02-07:00",
  "updatedBy": "admin",
  "lastUpdateDate": "2015-07-14T03:42:38.778-07:00",
  "consolidationInd": "1",
  "lastRowidSystem": "SYS0",
  "dirtyIndicator": "0",
  "interactionId": "72300000001000",
  "hubStateInd": "1",
  "label": "LLOYD,BOB",
  "partyType": "Person",
  "lastName": "LLOYD",
  "firstName": "BOB",
  "displayName": "BOB LLOYD",
  "preferredPhone": {
    "rowidObject": "164",
    "$original": {
      "rowidObject": "164"
    }
  },
  "$original": {
    "label": "DUNN,LLOYD",
    "lastName": "DUNN",
  }
}
```

```
        "firstName": "LLOYD",
        "displayName": "LLOYD DUNN"
    }
}
```

Promover

La API de REST de promoción promueve todos los cambios pendientes que se realicen en un registro en función del ID de interacción de la solicitud de cambio.

La API utiliza el método POST. Proporcione el ID de interacción como un parámetro de consulta.

URL de solicitud

La dirección URL para la promoción tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?action=promote&interactionId=<interaction ID>
```

Haga la siguiente solicitud HTTP POST a la URL para la promoción:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId of the root record>?action=promote&interactionId=<interaction ID>
```

Parámetro de consulta

El ID de interacción de los cambios pendientes es un parámetro obligatorio. La API utiliza el ID de interacción para buscar todos los registros relacionados con los cambios pendientes.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo promueve todos los cambios pendientes según el ID de interacción de la solicitud de cambio:

```
POST http://localhost:8080/cmxcs/localhost-orcl-DS_UI1/Person/1038246?
action=promote&interactionId=69120000294000
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo contiene el ID de fila del registro después de promover los cambios pendientes:

```
{
    Person: {
        rowidObject: "1038246"
    }
}
```

Eliminar pendiente

La API de REST de eliminación de pendientes elimina todos los cambios pendientes que realice en un registro en función del ID de interacción de la solicitud de cambio.

La API utiliza el método DELETE y devuelve el ID de fila del registro.

URL de solicitud

La URL para la eliminación de pendiente tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=deletePending&interactionId=<interaction ID>
```

Haga la siguiente solicitud DELETE a la URL para la eliminación de pendiente:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=deletePending&interactionId=<interaction ID>
```

Parámetro de consulta

Proporcione el ID de interacción de los cambios pendientes que desee eliminar.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina todos los cambios pendientes según el ID de interacción de la solicitud de cambio:

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
action=deletePending&interactionId=72300000001000
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo contiene el ID de fila del registro después de eliminar los cambios pendientes:

```
{
  Person: {
    rowidObject: "233"
  }
}
```

Vista previa de fusión

La API de REST de vista previa de fusión devuelve una vista previa de un registro raíz consolidado si fusiona dos o más registros raíz.

La API utiliza el método POST y acepta una lista de registros raíz y reemplazos a nivel de campo para devolver una vista previa del registro fusionado. El ID de fila del registro de destino es un parámetro obligatorio.

URL de solicitud

La URL para la vista previa de la fusión tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=previewMerge
```

El siguiente formato de la URL para la vista previa de la fusión especifica el número de niveles secundarios que se van a devolver:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=previewMerge&depth=2
```

Haga la siguiente solicitud HTTP POST a la URL para la vista previa de la fusión:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=previewMerge
```

Nota: En el cuerpo de la solicitud, añada la propiedad de las claves y especifique los registros raíz que desea fusionar con el registro de destino.

Para reemplazar las coincidencias para los registros secundarios, añada el parámetro contentMetadata:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge&contentMetadata=MERGE/<json/xml>
```

Nota: En el cuerpo de la solicitud, añada la propiedad overrides y especifique los reemplazos de fusión.

Para especificar el tipo de medio de los datos que desea enviar con la solicitud, añada el encabezado Content-Type:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?action=previewMerge
Content-Type: application/<json/xml>
```

Parámetros de consulta

El ID de fila del registro de destino es un parámetro obligatorio.

Puede utilizar los siguientes parámetros de consulta:

Parámetro	Descripción
contentMetadata	Metadatos para la vista previa de fusión. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- BVT. Devuelve información sobre el registro de referencias cruzadas prevalente y el ID de registro original.- MERGE. Devuelve información sobre cómo se han fusionado los registros descendientes.
depth	El número de niveles secundarios que se devolverán.
effectiveDate	Fecha en la que desea generar la vista previa.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Cuerpo de la solicitud

Antes de empezar, use la API de lectura de registros coincidentes para determinar qué registros coincidentes puede fusionar con el registro raíz original. Envíe la lista de registros en el cuerpo de la solicitud para la API de vista previa de fusión.

Puede reemplazar los valores de campo en el registro raíz. Por ejemplo, si ninguno de los registros raíz fusionados contiene la ortografía correcta del nombre, puede especificar el nombre en el cuerpo de la solicitud. Además, puede reemplazar el modo en el que se van a fusionar los registros secundarios mediante la eliminación de candidatos de fusión o la especificación de otros candidatos de fusión.

Use las siguiente propiedades en el cuerpo de la solicitud:

Propiedades/ elementos	Tipo	Descripción
keys	array	Obligatorio. Lista ordenada de los registros raíz similares que desea añadir a la fusión. Puede identificar los registros por el ID de fila o mediante una combinación del sistema de origen y la clave de origen.
overrides	object	Reemplace los valores de campo de un registro raíz y las coincidencias de los registros secundarios.
MERGE	object	Reemplace el modo de fusión de los registros secundarios. Añada el tipo de registro secundario al objeto <code>overrides</code> y, a continuación, añada el objeto <code>MERGE</code> .

En el siguiente ejemplo de código JSON se identifica un registro raíz que se va a fusionar con el registro raíz de destino:

```
{
  keys: [
    {
      rowid: "P2"
    }
  ]
}
```

En el siguiente código se muestra cómo reemplazar un campo del registro raíz de parte y cómo reemplazar los candidatos de fusión para los registros secundarios de teléfono:

```
{
  keys: [
    {
      rowid: "P2"
    }
  ]
  overrides: {
    Party: {
      rowidObject: "P1",
      firstName: "Serge", //override the value for the first name
      Telephone: { // override which Telephone child records to merge
        item:[
          {
            rowidObject: "T1",
            MERGE: {
              item: [ // to remove the original merge candidates, specify null
                null,
                null
              ],
              $original: {
                item: [
                  {key:{rowid: "T2"}},
                  {key:{rowid: "T3"}}
                ]
              }
            }
          }
        ],
        {
          rowidObject: "T4",
          MERGE: {
            item: [ // to add or change merge candidates, specify matched records
              {key:{rowid: "T2"}}
            ],
            $original: {
              item: [
                null
              ]
            }
          }
        }
      }
    }
  }
}
```

```

    }
  ]
}
}
}

```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve la vista previa de un registro consolidado:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=previewMerge

{
  keys: [
    {
      rowid: "2478246"
    },
    {
      rowid: "2478230"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la vista previa del registro consolidado:

```

{
  "Person": {
    "rowidObject": "2478245",
    "partyType": "Person",
    "lastName": "Smith",
    "firstName": "Charlie",
    "displayName": "ALICE SMITH"
  }
}

```

Fusión pendiente

La API de REST de fusión pendiente actualiza todas las tareas de fusión pendiente que se realizan en un registro en función del ID de interacción de la solicitud de cambio. Con Fusión pendiente, las operaciones de fusión se pueden aplazar hasta que el proceso de flujo de trabajo conceda la aprobación para todas las tareas de fusión.

La API utiliza el método POST y devuelve el ID de fila del registro.

URL de solicitud

La dirección URL para la fusión pendiente tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=PendingMerge&interactionId=<interaction ID>

```

Haga la siguiente solicitud HTTP POST a la URL para la fusión pendiente:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge

```


Parámetro de consulta

El ID de interacción de la fusión pendiente es un parámetro obligatorio.

Ejemplo de solicitud de API

El siguiente ejemplo de solicitud actualiza todas las tareas de fusión pendiente que están asociadas con el ID de interacción:

```
POST /Person/123?action=pendingMerge&interactionId=123
```

Ejemplo de respuesta de API

El siguiente ejemplo de respuesta contiene los ID de fila de los objetos base raíz afectados:

```
{
  keys: [{rowid: "456"}, {rowid: "789"}],
  overrides: {...}
}
```

Promover fusión

La API de REST para promover la fusión ejecuta todas las tareas de fusión pendiente que hay asociadas con el ID de interacción de la solicitud de cambio.

La API utiliza el método POST y devuelve el ID de fila del registro ganador.

URL de solicitud

La dirección URL para la promoción de la fusión tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid>?
action=PromoteMerge&interactionId=<interaction ID>
```

Haga la siguiente solicitud HTTP POST a la URL para promover la fusión:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge
```

Parámetro de consulta

El ID de interacción de la tarea de fusión pendiente es un parámetro obligatorio. La API utiliza el ID de interacción para buscar todas las tareas de fusión pendiente y ejecuta las fusiones.

Ejemplo de solicitud de API

El siguiente ejemplo de solicitud promueve todas las tareas de fusión pendiente que están asociadas con el ID de interacción:

```
POST /Person/123?action=promoteMerge&interactionId=123
```

Ejemplo de respuesta de API

El siguiente ejemplo de respuesta contiene los ID de fila de los registros después de promover las tareas de fusión pendiente:

```
POST /Person/123?action=promoteMerge&interactionId=123
```

Fusionar registros

La API de REST para la fusión de registros fusiona dos o más registros raíz para crear un único registro consolidado. El ID de fila del registro consolidado es el ID de fila del registro en el que se fusionan los demás registros.

La API utiliza el método POST. Puede especificar los reemplazos a nivel de campo para el registro fusionado en el cuerpo de la solicitud.

URL de solicitud

La URL para la fusión de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the record>?
action=Merge
```

Haga la siguiente solicitud HTTP POST a la URL para la fusión de registros:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowid of the
record>?action=Merge
Content-Type: application/<json/xml>
```

Parámetros de consulta

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

Cuerpo de la solicitud

Antes de empezar, use la API de vista previa de fusión para obtener una vista previa de los resultados de la fusión de los registros raíz seleccionados. Si considera que la vista previa es satisfactoria, use las mismas propiedades en el cuerpo de la solicitud para la API de fusión de registros.

Puede reemplazar los valores de campo en el registro raíz. Por ejemplo, si ninguno de los registros raíz fusionados contiene la ortografía correcta del nombre, puede especificar el nombre en el cuerpo de la solicitud. Además, puede reemplazar el modo en el que se van a fusionar los registros secundarios mediante la eliminación de candidatos de fusión o la especificación de otros candidatos de fusión.

Use las siguiente propiedades en el cuerpo de la solicitud:

Propiedades/elementos	Tipo	Descripción
keys	array	Obligatorio. Lista ordenada de los registros raíz similares que desea añadir a la fusión. Puede identificar los registros por el ID de fila o mediante una combinación del sistema de origen y la clave de origen.
overrides	object	Reemplace los valores de campo de un registro raíz y las coincidencias de los registros secundarios.
MERGE	object	Reemplace el modo de fusión de los registros secundarios. Añada el tipo de registro secundario al objeto <code>overrides</code> y, a continuación, añada el objeto <code>MERGE</code> .

En el siguiente ejemplo de código JSON se identifican dos registros raíz que se van a fusionar con el registro raíz de destino:

```
{
  keys: [
    {rowid: "2478246"},
    {rowid: "2478269"}
  ]
}
```

Para ver un ejemplo de uso de las propiedades `overrides` y `MERGE` con la API de fusión de registros, consulte el cuerpo de la solicitud para la API de vista previa de fusión.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo fusiona registros para formar un registro consolidado. La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-DS_UI1/Person/2478245?
action=merge&taskComment=Read my comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9
Content-Type: application/<json/xml>

{
  keys: [
    {
      rowid: "2478246"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el registro consolidado:

```
{
  "Person": {
    "link": [
      {
        "href": "http://localhost:8080/cmxc/cs/localhost-orcl-DS_UI1/Person/
2478245",
        "rel": "self"
      }
    ],
    "key": {
```

```

        "rowid": "2478245"
      },
      "rowidObject": "2478245"
    }
  }
}

```

Anular fusión de registros

La API de REST para la anulación de fusión de registros anula la fusión de un registro raíz en los registros raíz individuales que existían antes de que se fusionaran los registros.

La API utiliza el método POST.

URL de solicitud

La URL para la anulación de la fusión de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=unmerge
```

Haga la siguiente solicitud HTTP POST a la URL para la anulación de la fusión de registros:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=unmerge
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=unmerge
Content-Type: application/<json/xml>
```

Parámetros de consulta

La siguiente tabla muestra los parámetros que puede usar en la URL:

Parámetro	Descripción
taskComment	Añada un comentario a la tarea de flujo de trabajo que ha activado la API.
taskAttachments	Si los adjuntos de tarea están habilitados, adjunte un archivo a la tarea de flujo de trabajo que ha activado la API.

Cuerpo de la solicitud

Envíe la lista de registros cuya fusión del registro consolidado desea anular en el cuerpo de la solicitud. Utilice el ID de fila xref o el sistema de origen y la clave de fuente para especificar los registros.

Utilice la API de lectura de registros para obtener el ID de fila (rowId) de XREF del registro cuya fusión se va a anular. En la siguiente solicitud de ejemplo se recuperan los metadatos de XREF de un registro:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2638243?
contentMetadata=XREF
```

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo anula la fusión de un registro del registro consolidado. La solicitud añade un comentario y un adjunto a la tarea de flujo de trabajo que ha activado la API.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478248?
action=unmerge&taskComment=Read my
comment&taskAttachments=TEMP_SVR1.29OT8,TEMP_SVR1.29OT9

{
    rowid: "4880369"
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el registro cuya fusión ha anulado del registro consolidado:

```
{
  "Person": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/
2478249",
        "rel": "self"
      }
    ],
    "key": {
      "rowid": "2478249"
    },
    "rowidObject": "2478249"
  }
}
```

Leer una relación

La API de REST de lectura de relaciones devuelve los detalles de un registro de relación, como los tipos de parte, los valores de rowID y los nombres para mostrar de los dos registros.

La API usa el método GET.

URL de solicitud

La dirección URL de lectura de relaciones tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID of the relationship
record>
```

Haga la siguiente solicitud HTTP GET a la URL:

```
GET http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID of the
relationship record>
```

Parámetros de consulta

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
suppressLinks	Opcional. Indica si los vínculos entre elementos primarios y secundarios se pueden ver en la respuesta de la API. Establezca el parámetro en true para suprimir todos los vínculos entre elementos primarios y secundarios en la respuesta. Establezca el parámetro en false para no mostrar los vínculos en la respuesta de la API. El valor predeterminado es false.
depth	Opcional. El número de niveles secundarios que se devolverán.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se devuelven los detalles del registro de relación con el ID de fila 85, lo que se corresponde con el tipo de relación `ProductGroupProductGroupIsParentOfProductProducts`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85
```

En la siguiente solicitud de ejemplo se devuelven detalles con una profundidad igual a 2:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85?depth=2
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestran los detalles del registro de relación con el ID de fila 85, que es el tipo de relación `ProductGroupProductGroupIsParentOfProductProducts`:

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json?depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "85",
  "label": "Product Group Product Group is parent of Product Products",
  "rowidRelType": "9",
  "rowidHierarchy": "3",
  "from": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json?depth=2",
        "rel": "children"
      }
    ]
  }
}
```

```

    }
  ],
  "rowidObject": "86",
  "label": "Product Group",
  "productType": "Product Group",
  "productNumber": "Presenter2",
  "productName": "Presenter",
  "productDesc": "Presenter Family",
  "productTypeCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86.json",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86/productTypeCd.json?depth=2",
        "rel": "children"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/from/86/productTypeCd.json",
        "rel": "self"
      }
    ]
  },
  "productType": "Product Group"
}
},
"to": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85.json",
      "rel": "parent"
    }
  ]
},
"rowidObject": "66",
"label": "Products",
"productType": "Product",
"productNumber": "931307-0403",
"productName": "2.4 GHz Cordless Presenter",
"productDesc": "A cordless presenter to streamline your delivery.",
"productTypeCd": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66/productTypeCd.json",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66.json",
      "rel": "parent"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
ProductGroupProductGroupIsParentOfProductProducts/85/to/66/productTypeCd.json?depth=2",
      "rel": "children"
    }
  ]
}
],

```

```

        "productType": "Presenter"
    }
}
}

```

Crear una relación

La API de REST de creación de relaciones permite crear una relación entre los registros especificados. Para crear una relación entre registros, deben existir relaciones entre las entidades de negocio a las que pertenecen los registros. Por ejemplo, si desea especificar una relación entre Informática y John Smith, debe existir una relación entre las entidades de negocio Persona y Organización. Debe enviar los datos de la relación en el cuerpo de la solicitud.

La API usa los métodos PUT y POST.

URL de solicitud

La dirección URL de REST de creación de relaciones tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<relationship>?systemName=<name of the
source system>

```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Use la siguiente solicitud HTTP POST o PUT para la dirección URL:

```

POST http://<host>:<port>/<context>/<database ID>/<relationship>?systemName=<name of the
source system>

```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```

Content-Type: application/<json/xml>

```

Parámetros de URL

El nombre del sistema de origen es un parámetro obligatorio en la URL de solicitud.

Cuerpo de la solicitud

Envíe los datos para el registro de relación en el cuerpo de la solicitud REST. Utilice el formato JSON o el formato XML para enviar los datos. Especifique los valores de los parámetros obligatorios en el cuerpo de la solicitud.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo crea la relación `OrganizationEmploysPerson` entre una entidad de negocio de organización con el ID de fila 101 y una entidad de negocio de persona con el ID de fila 1101:

```

POST http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/OrganizationEmploysPerson?
systemName=SFA
Content-Type: application/json

{
  "from": {
    "rowidObject": "101"
  },
  "to": {
    "rowidObject": "1101"
  },
}

```



```

    "relName": "Documentation",
    "relDesc": "Writer"
  }

```

La relación `OrganizationEmploysPerson` define una relación de una entidad de negocio de organización con una entidad de negocio de persona. El elemento `from` especifica el registro en el que se origina la relación y el elemento `to` especifica el registro en el que finaliza la relación.

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de crear correctamente una relación entre una entidad de negocio de organización con el ID de fila 101 y una entidad de negocio de persona con el ID de fila 1101:

```

{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
      "sourceKey": "SVR1.1E7UW"
    }-
  }-
  "rowidObject": "414721"
  "from": {
    "key": {
      "rowid": "101 "
    }-
    "rowidObject": "101 "
  }-
  "to": {
    "key": {
      "rowid": "1101 "
    }-
  }-
  "rowidObject": "1101 "
}

```

Actualizar una relación

La API de REST de actualización de relaciones actualiza la relación entre dos registros. La API actualiza los atributos adicionales definidos para la relación.

La API usa los métodos POST y PUT.

URL de solicitud

La dirección URL de actualización de relaciones tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID>?systemName=<name of the source system>

```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Use la siguiente llamada a HTTP POST o PUT para la dirección URL:

```

http://<host>:<port>/<context>/<database ID>/<relationship>/<row ID>?systemName=<name of the source system>

```

Agregue el encabezado `Content-Type` para especificar el tipo de medio de los datos que desea enviar con la solicitud:

Cuerpo de la solicitud

Envíe las actualizaciones al registro de relación en el cuerpo de la solicitud. Utilice el formato JSON o el formato XML para enviar los datos. Especifique los valores de los parámetros obligatorios en el cuerpo de la solicitud.

Ejemplo de solicitud de API

La relación con el ID de fila 414721 es una relación OrganizationEmploysPerson entre una entidad de organización con el ID de fila 101 y una entidad de persona con el ID de fila 1101.

La siguiente solicitud de ejemplo actualiza el registro de relación con el ID de fila 414721:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson/414721?
systemName=SFA
Content-Type: application/json
{
  "from": {
    "rowidObject": "101"
  },
  "to": {
    "rowidObject": "1101"
  },
  "relName": "Development",
  "relDesc": "Software Engineer",
  "$original": {
    "relName": "Documentation",
    "relDesc": "Writer"
  }
}
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo se recibe una vez que se ha actualizado correctamente la relación con el ID de fila 414721:

```
{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
      "sourceKey": "SVR1.1E7UW"
    }-
    "rowidObject": "414721"
    "from": {
      "key": {
        "rowid": "101"
      }-
      "rowidObject": "101"
    }-
    "to": {
      "key": {
        "rowid": "1101"
      }-
      "rowidObject": "1101"
    }-
  }-
}
```

Eliminar una relación

La API de REST de eliminación de relaciones elimina la relación entre dos registros.

La API usa el método DELETE.

URL de solicitud

La dirección URL de eliminación de relaciones tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<relationship>/<rowID of the relationship record>?systemName=<name of the source system>
```

Nota: El nombre del sistema de origen es un parámetro obligatorio en la URL.

Haga la siguiente solicitud HTTP DELETE a la dirección URL de eliminación:

```
DELETE http://<host>:<port>/<context>/<database ID>/<relationship>/<rowID of the relationship record>?systemName=<name of the source system>
```

Parámetro de consulta

El nombre del sistema de origen es un parámetro de URL obligatorio. Use el parámetro `systemName` para especificar el sistema de origen.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se elimina un registro de relación con el ID de fila 414721:

```
DELETE http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/OrganizationEmploysPerson/414721?systemName=SFA
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra la respuesta después de eliminar correctamente el registro de relación con el ID de fila 414721:

```
{
  "OrganizationEmploysPerson": {
    "key": {
      "rowid": "414721"
    }-
  "rowidObject": "414721"
  }-
}
```

Obtener registros relacionados

La API de REST de obtención de registros relacionados devuelve una lista de registros relacionados con un registro raíz especificado y basado en las relaciones que ha configurado. La API devuelve también los detalles de las relaciones.

La API usa el método GET.

URL de solicitud

La URL para la obtención de registros relacionados tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=related
```

Haga la siguiente solicitud HTTP GET a la URL para la obtención de registros relacionados:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=related
```

Parámetros de consulta

Puede anexar los parámetros de consulta a la dirección URL de la solicitud.

En la siguiente tabla se indican los parámetros de consulta que puede usar:

Parámetro	Descripción
recordsToReturn	Número de registros para los elementos secundarios <i>many</i> que desea leer.
searchToken	Token de búsqueda para obtener las páginas siguientes del conjunto de resultados.
returnTotal	Devuelve el número de registros en el conjunto de resultados. Se establece en <i>true</i> para obtener el número de registros de la lista de resultados. El valor predeterminado es <i>false</i> .

Parámetros de filtro

Puede anexar parámetros a la dirección URL para filtrar los registros relacionados.

En la siguiente tabla se indican los parámetros de filtro que puede usar:

Parámetro	Descripción
recordStates	Lista de estados de registro separados por comas que desea recuperar. Los estados de registro admitidos son ACTIVE, PENDING y DELETED. El valor predeterminado es ACTIVE. Por ejemplo, la consulta <code>/Party/123?action=related&recordStates=ACTIVE,PENDING</code> devuelve registros activos o pendientes.
entityLabel	Etiqueta de la entidad.
relationshipLabel	Etiqueta de la relación.
entityType	Lista de tipos de entidad separados por comas. Por ejemplo, la lista <code>entityType=Person,Organization</code> devuelve registros relacionados de los tipos de entidad <i>Person</i> y <i>Organization</i> .
relationshipType	Lista de tipos de relación separados por comas. Por ejemplo, la lista <code>relationshipType=Employee,Employer</code> devuelve registros relacionados de los tipos de relación <i>Employee</i> y <i>Employer</i> .

Nota: Si especifica varias condiciones de filtro, el resultado contiene todos los registros que cumplen la condición AND.

Cuerpo de la respuesta

El cuerpo de la respuesta contiene la lista de registros relacionados, los detalles de las relaciones y los registros relacionados, y un token de búsqueda. Utilice el token de búsqueda para obtener las páginas posteriores del resultado.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se obtienen los registros relacionados y las relaciones configuradas para la entidad de negocio de organización con el ID de fila 101:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Organization/101?action=related
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los registros relacionados y las relaciones para el tipo de entidad de negocio de organización con el ID de fila 101:

```
{
  "link": [],
  "firstRecord": 1,
  "pageSize": 10,
  "searchToken": "SVR1.1H7YB",
  "relatedEntity": [
    {
      "businessEntity": {
        "SecurePerson": {
          "link": [
            {
              "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_ui1/SecurePerson/1101",
              "rel": "self"
            }
          ]
        },
        "rowidObject": "1101",
        "creator": "admin",
        "createDate": "2008-11-11T21:22:20-08:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2012-03-29T19:03:19-07:00",
        "consolidationInd": "1",
        "lastRowidSystem": "SYS0",
        "dirtyIndicator": "0",
        "interactionId": "20003000",
        "hubStateInd": "1",
        "partyType": "Person",
        "lastName": "Obama",
        "firstName": "Barack"
      }
    },
    {
      "entityLabel": "Obama,Barack",
      "relationshipLabel": "Organization employes SecurePerson",
      "relationship": {
        "rowidObject": "414721",
        "creator": "admin",
        "createDate": "2016-10-17T01:58:12.436-07:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-19T01:40:28.830-07:00",
        "consolidationInd": "4",
        "lastRowidSystem": "SFA",
        "interactionId": "1476866426786",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
      },
      "entityType": "SecurePerson",
      "relationshipType": "OrganizationEmployesSecurePerson"
    }
  ],
  {
    "businessEntity": {
      "SecurePerson": {
        "link": [
          {
            "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_ui1/SecurePerson/114",

```

```

        "rel": "self"
    }
},
"rowidObject": "114",
"creator": "admin",
"createDate": "2008-08-11T23:00:55-07:00",
"updatedBy": "Admin",
"lastUpdateDate": "2008-08-12T02:59:17-07:00",
"consolidationInd": "1",
"lastRowidSystem": "Legacy",
"dirtyIndicator": "0",
"hubStateInd": "1",
"partyType": "Person",
"lastName": "HERNANDEZ",
"displayName": "ALEJANDRO HERNANDEZ",
"firstName": "ALEJANDRO"
},
"entityLabel": "HERNANDEZ,ALEJANDRO",
"relationshipLabel": "Organization employes SecurePerson",
"relationship": {
    "rowidObject": "434721",
    "creator": "admin",
    "createDate": "2016-10-19T01:49:03.415-07:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-19T01:49:03.415-07:00",
    "consolidationInd": "4",
    "lastRowidSystem": "SFA",
    "hubStateInd": "1",
    "rowidRelType": "101",
    "rowidHierarchy": "1",
    "relName": "Documentation",
    "relDesc": "Writer"
},
"entityType": "SecurePerson",
"relationshipType": "OrganizationEmployesSecurePerson"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://10.21.43.42:8080/cmxcsllocalhost-orcl-ds_ui1/Person/1101",
                    "rel": "self"
                }
            ]
        }
    },
    "rowidObject": "1101",
    "creator": "admin",
    "createDate": "2008-11-11T21:22:20-08:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2012-03-29T19:03:19-07:00",
    "consolidationInd": "1",
    "lastRowidSystem": "SYS0",
    "dirtyIndicator": "0",
    "interactionId": "20003000",
    "hubStateInd": "1",
    "partyType": "Person",
    "lastName": "Obama",
    "firstName": "Barack"
},
"entityLabel": "Obama,Barack",
"relationshipLabel": "Organization employes Person",
"relationship": {
    "rowidObject": "414721",
    "creator": "admin",
    "createDate": "2016-10-17T01:58:12.436-07:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-19T01:40:28.830-07:00",
    "consolidationInd": "4",
    "lastRowidSystem": "SFA",

```

```

        "interactionId": "1476866426786",
        "hubStateInd": "1",
        "rowidRelType": "101",
        "rowidHierarchy": "1",
        "relName": "Documentation",
        "relDesc": "Writer"
    },
    "entityType": "Person",
    "relationshipType": "OrganizationEmployesPerson"
},
{
    "businessEntity": {
        "Person": {
            "link": [
                {
                    "href": "http://10.21.43.42:8080/cmx/cs/localhost-orcl-ds_uil/Person/114",
                    "rel": "self"
                }
            ]
        },
        "rowidObject": "114",
        "creator": "admin",
        "createDate": "2008-08-11T23:00:55-07:00",
        "updatedBy": "Admin",
        "lastUpdateDate": "2008-08-12T02:59:17-07:00",
        "consolidationInd": "1",
        "lastRowidSystem": "Legacy",
        "dirtyIndicator": "0",
        "hubStateInd": "1",
        "partyType": "Person",
        "lastName": "HERNANDEZ",
        "displayName": "ALEJANDRO HERNANDEZ",
        "statusCd": "A",
        "firstName": "ALEJANDRO"
    }
},
"entityLabel": "HERNANDEZ,ALEJANDRO",
"relationshipLabel": "Organization employes Person",
"relationship": {
    "rowidObject": "434721",
    "creator": "admin",
    "createDate": "2016-10-19T01:49:03.415-07:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-19T01:49:03.415-07:00",
    "consolidationInd": "4",
    "lastRowidSystem": "SFA",
    "hubStateInd": "1",
    "rowidRelType": "101",
    "rowidHierarchy": "1",
    "relName": "Documentation",
    "relDesc": "Writer"
},
"entityType": "Person",
"relationshipType": "OrganizationEmployesPerson"
}
]
}

```

Leer registros coincidentes

La API de REST para la lectura de registros coincidentes devuelve registros que coinciden con un registro raíz específico. Puede revisar la lista de registros para determinar qué registros puede fusionar con el registro raíz original. Con la API para fusionar registros podrá fusionar los registros.

La API utiliza el método GET.

URL de solicitud

La URL para la lectura de registros coincidentes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Haga la siguiente solicitud HTTP GET a la URL para la lectura de registros coincidentes:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Cuerpo de la respuesta

El cuerpo de la respuesta contiene el número de registros que coinciden con el registro especificado, los detalles de los registros coincidentes y un token de búsqueda. Utilice el token de búsqueda para obtener las páginas posteriores del resultado de coincidencias.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo busca la entidad de negocio para los registros que coincidan con un registro:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/1038245?action=matched
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra los detalles del registro que coincide con el registro especificado:

```
{
  "firstRecord": 1,
  "recordCount": 1,
  "pageSize": 10,
  "searchToken": "SVR1.AU5HE",
  "matchedEntity": [
    {
      "businessEntity": {
        "Person": {
          "link": [
            {
              "href": "http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/
Person/1038246",
              "rel": "self"
            }
          ],
          "rowidObject": "1038246",
          "creator": "admin",
          "createDate": "2008-08-12T02:15:02-07:00",
          "updatedBy": "Admin",
          "lastUpdateDate": "2008-08-12T02:59:17-07:00",
          "consolidationInd": "1",
          "lastRowidSystem": "SFA",
          "dirtyIndicator": "0",
          "hubStateInd": "1",
          "partyType": "Person",
          "lastName": "BATES",
          "firstName": "DAISY",
          "displayName": "DAISY BATES"
        }
      },
      "matchRule": "PUT"
    }
  ]
}
```


Actualizar registros coincidentes

La API de REST para la actualización de registros coincidentes crea o actualiza un registro en la tabla de coincidencias. La tabla de coincidencias contiene los pares de registros coincidentes de una entidad de negocio después de ejecutar un proceso de coincidencia en la entidad de negocio. Utilice la API para agregar los registros que son aptos para una fusión con el registro especificado.

La API utiliza el método PUT.

URL de solicitud

La URL para la actualización de los registros coincidentes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Haga la siguiente solicitud HTTP PUT a la URL para la actualización de registros coincidentes:

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
PUT http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Envíe la lista de registros que coinciden con el registro especificado en el cuerpo de la solicitud. Utilice el ID de fila o el sistema de origen y la clave de fuente para especificar los registros.

Ejemplo de solicitud de API

El siguiente ejemplo agrega un registro en la tabla de coincidencias:

```
PUT http://localhost:8080/cmx/cs/localhost-ORCL-DS_UI1/Person/1038245?action=matched
{
  keys: [
    {
      rowid: "1038246"
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve una respuesta de 200 OK al crear correctamente un registro en la tabla de coincidencias. El cuerpo de la respuesta está vacío.

Eliminar registros coincidentes

La API de REST para la eliminación de registros coincidentes elimina los registros coincidentes que estén asociados a un registro raíz en la tabla de coincidencia.

La API utiliza el método DELETE.

URL de solicitud

La URL para la eliminación de registros coincidentes tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?action=matched
```

Haga la siguiente solicitud HTTP DELETE a la URL para la eliminación de registros coincidentes:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=matched
```

Agregue el encabezado Content-Type para especificar el tipo de medio de los datos que desea enviar con la solicitud:

```
DELETE http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=matched
Content-Type: application/<json/xml>
```

Cuerpo de la solicitud

Envíe la lista de registros que desea eliminar de la tabla de coincidencia en el cuerpo de la solicitud.

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo elimina un registro que coincida con el registro raíz especificado de la tabla de coincidencia:

```
DELETE http://localhost:8080/cmxcsllocalhost-orcl-DS_UI1/Person/1038245?action=matched
{
  keys: [
    {
      rowid: "1038246"
    }
  ]
}
```

Ejemplo de respuesta de API

La API devuelve una respuesta de 200 OK al eliminar correctamente un registro de la tabla de coincidencias. El cuerpo de la respuesta está vacío.

Obtener eventos de historial de registros

La API de REST de obtención de eventos de historial de registros devuelve una lista de eventos de historial o grupos de eventos de historial asociados a un registro. Envíe el ID del registro en el cuerpo de la solicitud.

La API usa el método GET para devolver los siguientes datos para cada grupo de eventos de historial:

- Fechas de inicio y finalización del grupo
- Número de eventos del grupo

La API devuelve los siguientes datos para cada evento de historial:

- ID del evento de historial
- Fecha del cambio
- Usuario que ha realizado el cambio
- Lista de tablas de historial afectadas por el cambio
- Lista de nodos de registro afectados por el cambio

URL de solicitud

La dirección URL de obtención de eventos de historial de registros tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=listHistoryEvents
```

Haga la siguiente solicitud HTTP GET a la dirección URL de obtención de eventos de historial de registros:

```
GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=listHistoryEvents
```

Parámetros de consulta

El ID del registro es un parámetro obligatorio. La API usa el ID de registro para buscar todos los eventos de historial relacionados.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
startDate y endDate	Opcional. Intervalo de fechas para el que desea recuperar los datos. Si especifica un intervalo de fechas, la respuesta contiene solo eventos de este intervalo.
granularity	Opcional. Nivel de detalle para agrupar eventos de historial. Si se especifica, la respuesta agrupa los eventos de historial. En caso contrario, la respuesta no agrupa los eventos de historial. Use uno de los siguientes valores: <ul style="list-style-type: none">- YEAR- QUARTER- MONTH- WEEK- DAY- HOUR- MINUTE- AUTO
recordStates	Opcional. Estados de registro devueltos en la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- ACTIVE- PENDING- DELETED
contentMetadata	Opcional. Metadatos para la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none">- XREF- PENDING_XREF- DELETED_XREF- HISTORY- MATCH- BVT- TRUST
children	Opcional. Lista separada por comas de los nombres de nodos secundarios. Si se especifica, la respuesta contiene los nombres de los nodos secundarios.
order	Opcional. Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.

Parámetro	Descripción
fields	Opcional. Lista de campos de entidad de negocio separados por comas. Si se especifica, la respuesta solo contiene los campos indicados.
filter	Opcional. Expresión de filtro.
depth	Opcional. El número de niveles secundarios que se devolverán.
recordsToReturn	Opcional. Especifica el número de filas que se van a devolver.
searchToken	Opcional. Especifica el token de búsqueda que se devuelve con la solicitud anterior.
returnTotal	Opcional. Si se establece en <code>true</code> , devuelve el número de registros en el resultado. El valor predeterminado es <code>false</code> .
firstRecord	Opcional. Especifica la primera fila del resultado.
changeType	Opcional. Especifica los tipos de cambio devueltos en el resultado. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none"> - BO - XREF - BVT - MERGE - MERGE_AS_SOURCE - MERGE_AS_TARGET - UNMERGE_AS_SOURCE - UNMERGE_AS_TARGET

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Ejemplo de solicitud de API

La siguiente solicitud de ejemplo devuelve todas las fusiones, agrupadas por año, para un registro desde el 1 de enero de 2000:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123?
action=listHistoryEvents&startDate=2010-01-01&granularity=YEAR&depth=2&changeType=MERGE
```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo indica las fusiones para el registro especificado desde el 1 de enero de 2000:

```
{
  firstRecord: 1,
  recordCount: 2
  item: [
    {
      link: [ // you can use links directly to get event list
        {rel: "events", href: "/Person/123?
action=listHistoryEvents&startDate=2000-01-01&endDate=2001-01-01&depth=2&changeType=MERGE
"}
      ],
      startDate: "2000-01-01",
```

```

        endDate: "2001-01-01",
        eventCount: 123
    },
    // no events in 2001, 2002, ... 2009
    {
        link: [
            {rel: "events", href: "/Person/123?
action=listHistoryEvents&startDate=2010-01-01&endDate=2011-01-01&depth=2&changeType=MERGE
"}
        ],
        startDate: "2010-01-01",
        endDate: "2011-01-01",
        eventCount: 23
    }
    // no events in 2011, ..., 2016
]
}

```

Obtener detalles de evento

La API de REST de obtención de detalles de evento devuelve los detalles de un evento de historial específico asociado a un registro. La API devuelve detalles como el tipo de cambio realizado y los valores antes y después del cambio. Envíe el ID de registro y el ID de evento de historial en el cuerpo de la solicitud.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de detalles de evento tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=getHistoryEventDetails

```

Use la siguiente solicitud HTTP GET para la dirección URL de obtención de detalles de evento:

```

GET http://<host>:<port>/<context>/<database ID>/<business entity>/<rowId>?
action=getHistoryEventDetails

```

Parámetros de consulta

Antes de empezar, use la API de obtención de eventos de historial de registros para indicar los eventos del historial que están asociados a un registro. En los resultados que se devuelven, use el ID de evento de historial y el ID de registro como parámetros de consulta.

En la tabla siguiente se muestran los parámetros de la consulta:

Parámetro	Descripción
eventID	Obligatorio. La API de obtención de eventos de historial de registros devuelve los ID de evento de los eventos del historial.
recordStates	Opcional. Estados de registro devueltos en la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: <ul style="list-style-type: none"> - ACTIVE - PENDING - DELETED

Parámetro	Descripción
contentMetadata	Opcional. Metadatos para la lista de eventos del historial. Proporcione una lista de valores separados por comas. Puede usar los siguientes valores: - XREF - PENDING_XREF - DELETED_XREF - HISTORY - MATCH - BVT - TRUST
children	Opcional. Lista separada por comas de los nombres de nodos secundarios. Si se especifica, la respuesta contiene los nombres de los nodos secundarios.
order	Opcional. Lista de nombres de campo separados por comas con el prefijo opcional + o -. El prefijo + indica la ordenación ascendente de los resultados y el prefijo - indica la ordenación descendente de los resultados. El valor predeterminado es +. Si especifica más de un parámetro, el conjunto de resultados se ordena por el primer parámetro de la lista, seguido del segundo.
fields	Opcional. Lista de campos de entidad de negocio separados por comas. Si se especifica, la respuesta solo contiene los campos indicados.
filter	Opcional. Expresión de filtro.
depth	Opcional. El número de niveles secundarios que se devolverán.
recordsToReturn	Opcional. Especifica el número de filas que se van a devolver.
searchToken	Opcional. Especifica el token de búsqueda que se devuelve con la solicitud anterior.
returnTotal	Opcional. Si se establece en <code>true</code> , devuelve el número de registros en el resultado. El valor predeterminado es <code>false</code> .
firstRecord	Opcional. Especifica la primera fila del resultado.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se devuelve información para un evento de historial:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/123?
action=getHistoryEventDetails&eventId=2016-07-14T02%3A01%3A24.529%2B0000
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los detalles del evento especificado:

```
{
  "eventId": "2016-07-14T02:01:24.529+0000",
  "eventDate": "2016-07-14T02:01:24.529Z",
  "user": "admin",
  "changeType": [
    "BVT",
    "BO",
    "UNMERGE_AS_TARGET"
  ],
  "businessEntity": {
```

```

    "Person": {
      "rowidObject": "438243",
      "creator": "datasteward1",
      "createDate": "2016-07-08T20:42:47.402Z",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-07-14T01:42:50.841Z",
      "consolidationInd": 1,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "BE,AC",
      "partyType": "Person",
      "lastName": "BE",
      "displayName": "AC BE",
      "firstName": "AC"
    }
  }
}

```

Obtener metadatos de DaaS

La API de REST de obtención de metadatos de DaaS devuelve información sobre un proveedor de DaaS, como el nombre, el tipo, la entidad de negocio con la que trabaja y la lista de campos obligatorios.

La API usa el método GET.

URL de solicitud

La dirección URL de obtención de metadatos de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/meta/daas/<providerName>
```

Use la siguiente solicitud HTTP GET para la dirección URL de obtención de metadatos de DaaS:

```
GET http://<host>:<port>/<context>/<database ID>/meta/daas/<providerName>
```

Parámetro de consulta

El parámetro `providerName` es obligatorio. El parámetro es el nombre del proveedor de DaaS configurado.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se devuelve la información de los metadatos del proveedor de DaaS `dcp`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/daas/dcp
```

En la siguiente solicitud de ejemplo se devuelve la información de los metadatos del proveedor de DaaS `ondemand`:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/meta/daas/ondemand
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestra la información de los metadatos del proveedor de DaaS `dcp` en formato JSON:

```

{
  "providerName": "dcp",
  "providerType": "READ",
  "businessEntity": "Organization",
  "systemName": "SFA",
  "requiredFields": [
    "dunsNumber"
  ]
}

```

En el siguiente ejemplo se muestra la información de los metadatos del proveedor de DaaS `ondemand` en formato JSON:

```
{
  providerName: "ondemand",
  providerType: "SEARCH",
  businessEntity: "Organization",
  systemName: "SFA",
  requiredFields: [
    "displayName"
  ]
}
```

Búsqueda de DaaS

La API de REST de búsqueda de DaaS usa determinados campos de entrada de una entidad de negocio para llamar a un servicio de DaaS externo y transforma la respuesta en una lista de registros.

La API usa el método POST.

URL de solicitud

La dirección URL de búsqueda de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/search/<daas provider>
```

Use la siguiente solicitud HTTP POST para la dirección URL de búsqueda de DaaS:

```
POST http://<host>:<port>/<context>/<database ID>/daas/search/<daas provider>
```

Nota: En el cuerpo de la solicitud, especifique los detalles de la entidad de negocio con el campo obligatorio.

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o un elemento JSON del tipo `Organization` u `OrganizationView` en el espacio de nombres `urn:co-ors.informatica.mdm`.

Ejemplo de solicitud de API

El siguiente ejemplo es una solicitud para el proveedor de DaaS `ondemand` para buscar la entidad de negocio `organization` con el nombre para mostrar `INFA`:

```
POST http://localhost:8080/cmxcsllocalhost-orcl-MDM_SAMPLE/daas/search/ondemand
{
  "Organization": {
    "displayname": "INFA"
  }
}
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los resultados de una búsqueda que el proveedor de DaaS devuelve para la entidad de negocio de organización con el nombre para mostrar `INFA`:

```
{
  "link": [],
  "item": [
    {
      "businessEntity": {
        "Organization": {
          "displayName": "INFA LAB INC",
          "dunsNumber": "001352574",
          "TelephoneNumbers": {
            "link": [],

```



```

        "item": [
            {
                "phoneNum": "9736252265"
            }
        ]
    },
    "Addresses": {
        "link": [],
        "item": [
            {
                "Address": {
                    "cityName": "ROCKAWAY",
                    "addressLine1": "11 WALL ST",
                    "postalCd": "07866",
                    "stateCd": {
                        "stateAbbreviation": "NJ"
                    }
                }
            }
        ]
    }
},
"label": "001352574-INFA LAB INC",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "007431013",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "5629019971"
                    }
                ]
            }
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "LONG BEACH",
                        "addressLine1": "3569 GARDENIA AVE",
                        "postalCd": "90807",
                        "stateCd": {
                            "stateAbbreviation": "CA"
                        }
                    }
                }
            ]
        }
    }
},
"label": "007431013-INFA INC",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "020591086",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "7186248777"
                    }
                ]
            }
        }
    }
}

```

```

    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "BROOKLYN",
            "addressLine1": "16 COURT ST STE 2004",
            "postalCd": "11241",
            "stateCd": {
              "stateAbbreviation": "NY"
            }
          }
        }
      ]
    }
  },
  "label": "020591086-INFA INC",
  "systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "604057286",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "8473580802"
          }
        ]
      }
    },
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "PALATINE",
            "addressLine1": "THE HARRIS BANK BLDG STE 614,800E NW HWY",
            "postalCd": "60074",
            "stateCd": {
              "stateAbbreviation": "IL"
            }
          }
        }
      ]
    }
  },
  "label": "604057286-INFA INC",
  "systemName": "SFA"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "032785606",
      "TelephoneNumbers": {
        "link": [],
        "item": [
          {
            "phoneNum": "5629019971"
          }
        ]
      }
    },
    "Addresses": {
      "link": [],
      "item": [
        {

```

```

        "Address": {
            "cityName": "SIMI VALLEY",
            "addressLine1": "3962 HEMWAY CT",
            "postalCd": "93063",
            "stateCd": {
                "stateAbbreviation": "CA"
            }
        }
    }
}
},
"label": "032785606-INFA INC",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA",
            "dunsNumber": "045228877",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "3304410033"
                    }
                ]
            }
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "NORTON",
                        "addressLine1": "4725 ROCK CUT RD",
                        "postalCd": "44203",
                        "stateCd": {
                            "stateAbbreviation": "OH"
                        }
                    }
                }
            ]
        }
    }
},
"label": "045228877-INFA",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "609028209",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "9084394655"
                    }
                ]
            }
        },
        "Addresses": {
            "link": [],
            "item": [
                {
                    "Address": {
                        "cityName": "CALIFON",
                        "addressLine1": "21 FAIRMOUNT RD W",
                        "postalCd": "07830",
                        "stateCd": {

```

```

        "stateAbbreviation": "NJ"
    }
}
]
}
},
"label": "609028209-INFA INC",
"systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "195271796",
            "TelephoneNumbers": {
                "link": [],
                "item": [
                    {
                        "phoneNum": "7137824181"
                    }
                ]
            },
            "Addresses": {
                "link": [],
                "item": [
                    {
                        "Address": {
                            "cityName": "HOUSTON",
                            "addressLine1": "1800 AUGUSTA DR STE 200",
                            "postalCd": "77057",
                            "stateCd": {
                                "stateAbbreviation": "TX"
                            }
                        }
                    }
                ]
            }
        }
    },
    "label": "195271796-INFA INC",
    "systemName": "SFA"
},
{
    "businessEntity": {
        "Organization": {
            "displayName": "INFA INC",
            "dunsNumber": "098605830",
            "Addresses": {
                "link": [],
                "item": [
                    {
                        "Address": {
                            "cityName": "BELLEFLOWER",
                            "postalCd": "90707",
                            "stateCd": {
                                "stateAbbreviation": "CA"
                            }
                        }
                    }
                ]
            }
        }
    },
    "label": "098605830-INFA INC",
    "systemName": "SFA"
}
]
}

```

Lectura de DaaS

La API de REST de lectura de DaaS usa determinados campos de una entidad de negocio para solicitar un servicio de DaaS externo y transforma la respuesta en un registro completo.

La API usa el método POST. Especifique el campo obligatorio en la solicitud para el proveedor de DaaS.

URL de solicitud

La dirección URL de lectura de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/read/<daas provider>
```

Use la siguiente solicitud POST para la dirección URL:

```
POST http://<host>:<port>/<context>/<database ID>/daas/read/<daas provider>
```

Nota: En el cuerpo de la solicitud, especifique los detalles del registro con el campo obligatorio.

Parámetro de consulta

En la siguiente tabla se indica el parámetro de consulta que puede usar:

Parámetro	Descripción
logChanges	Opcional. Si se establece en true, el registro resultante incluye el resumen de los cambios del objeto de datos de servicio (SDO), el cual se pasa al servicio de entidad de negocio de escritura. El valor predeterminado es false.

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o un elemento JSON del tipo `OrganizationView` en el espacio de nombres `urn:coors.informatica.mdm`.

Ejemplo de solicitud de API

El siguiente ejemplo es una solicitud para el proveedor de DaaS `ondemand` para buscar la entidad de negocio `organization` con el nombre para mostrar `INFA`:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/search/ondemand
{
  "Organization": {
    "displayname": "INFA"
  }
}
```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran los detalles que el proveedor de DaaS ha devuelto para la organización con el número D-U-N-S `001352574`:

```
{
  "Organization": {
    "displayName": "Infa Lab Inc",
    "dunsNumber": "001352574",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "phoneNum": "09736250550"
        }
      ]
    }
  }
}
```

```

    ],
    "Addresses": {
      "link": [],
      "item": [
        {
          "Address": {
            "cityName": "Rockaway",
            "addressLine1": "11 WALL ST"
          }
        }
      ]
    }
  }
}

```

WriteMerge

La API de REST WriteMerge acepta una lista de registros recuperados de un proveedor de DaaS, los conserva en XREF independientes con el sistema de origen correspondiente y los fusiona en un solo registro. Todas las XREF pertenecen al mismo registro.

La API usa el método POST.

URL de solicitud

La dirección URL de WriteMerge tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/write-merge/<business entity name>
```

Use la siguiente solicitud HTTP POST para la dirección URL de WriteMerge:

```
POST http://<host>:<port>/<context>/<database ID>/daas/write-merge/<business entity name>
```

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o JSON del tipo `DaaSEntity.Pager` en el espacio de nombres `urn:cobase.informatica.mdm`.

Encabezado de respuesta

Si la respuesta es satisfactoria, la API devuelve `interactionId` y `processId` en el encabezado de la respuesta.

Ejemplo de solicitud de API

En la siguiente solicitud de ejemplo se usa el resultado de una búsqueda de DaaS como entrada para crear un registro del tipo de entidad de negocio de organización:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/write-merge/Organization
{
  "item": [
    {
      "businessEntity": {
        "Organization": {
          "displayName": "INFA LAB INC",
          "dunsNumber": "001352574",
          "TelephoneNumbers": {
            "item": [
              {
                "phoneNum": "9736252265"
              }
            ]
          }
        }
      }
    }
  ]
}

```

```

    }
  },
  "systemName": "DNB"
},
{
  "businessEntity": {
    "Organization": {
      "displayName": "INFA INC",
      "dunsNumber": "007431013",
      "TelephoneNumbers": {
        "item": [
          {
            "phoneNum": "5629019971"
          }
        ]
      }
    }
  }
},
"systemName": "Admin"
}
]
}

```

Ejemplo de respuesta de API

La siguiente respuesta de ejemplo muestra el encabezado y el cuerpo de la respuesta después de fusionar correctamente la lista de registros en un solo registro:

```

{
  "Organization": {
    "key": {
      "rowid": "121921",
      "sourceKey": "140000000000"
    },
    "rowidObject": "121921",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "21721",
            "sourceKey": "140000001000"
          },
          "rowidObject": "21721"
        }
      ]
    }
  }
}

```

Importación de DaaS

La API de REST de importación de DaaS acepta datos en formato XML y los convierte en un registro.

La API usa el método POST.

URL de solicitud

La dirección URL de importación de datos de DaaS tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/daas/import/
FamilyTreeMemberOrganizationToOrgView

```

Use la siguiente solicitud HTTP POST para la dirección URL de importación de datos de DaaS:

```

POST http://<host>:<port>/<context>/<database ID>/daas/import/
FamilyTreeMemberOrganizationToOrgView

```

Parámetros de consulta

El nombre del sistema de origen es un parámetro de obligatorio.

En la siguiente tabla se indican los parámetros que puede usar en la solicitud:

Parámetro	Descripción
systemName	Obligatorio. Nombre del sistema de origen que realiza el cambio en los datos.
interactionId	Opcional. ID de interacción que se asigna a todos los cambios. Normalmente, el concentrador genera el ID al crear un cambio pendiente como resultado de una llamada.
effectivePeriod	Opcional. Contiene la fecha de inicio y la fecha de finalización. Especifica el periodo de tiempo en el que el registro es efectivo. Especifique estos parámetros para un registro habilitado para la línea temporal.
validateOnly	Opcional. Si se establece en TRUE, solo se aplican las reglas de validación al registro modificado y los cambios no son persistentes.
recordState	Opcional. Estado del concentrador para los registros creados o actualizados. Los estados de registro admitidos son ACTIVE y PENDING.
processId	Opcional. ID del proceso de flujo de trabajo que contiene la tarea. Si un flujo de trabajo se inicia como un resultado de la llamada de servicio, el parámetro contiene el identificador del proceso de flujo de trabajo iniciado.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o JSON del tipo

`DaaSChangeFamilyTreeMemberOrganizationToOrgView` en el espacio de nombres `urn:co-ors.informatica.mdm`.

Encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (`interactionId`) y el ID de proceso (`processId`) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```
BES-interactionId: 72200000242000  
BES-processId: 15948
```

Ejemplo de solicitud de API

La solicitud siempre tiene el formato XML.

En la siguiente solicitud de ejemplo se muestran los datos XML del tipo ChildAssociation del espacio de nombres de vinculación:

```

POST http://localhost:8080/cmxcsllocalhost-orcl-MDM_SAMPLE/daas/import/linkage2org?
systemName=Admin
<FamilyTreeMemberOrganization xmlns="http://services.dnb.com/LinkageServiceV2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ChildAssociation">
  <AssociationTypeText>ParentSubsidiary</AssociationTypeText>
  <OrganizationName>
    <OrganizationPrimaryName>
      <OrganizationName>INFORMATICA JAPAN K.K.</OrganizationName>
    </OrganizationPrimaryName>
  </OrganizationName>
  <SubjectHeader>
    <DUNSNumber>697557825</DUNSNumber>
  </SubjectHeader>
  <Location>
    <PrimaryAddress>
      <StreetAddressLine>
        <LineText>2-5-1, ATAGO</LineText>
      </StreetAddressLine>
      <StreetAddressLine>
        <LineText>ATAGO GREEN HILLS MORI TOWER 26F.</LineText>
      </StreetAddressLine>
      <PrimaryTownName>MINATO-KU</PrimaryTownName>
      <CountryISOAlpha2Code>JP</CountryISOAlpha2Code>
      <TerritoryAbbreviatedName>TKY</TerritoryAbbreviatedName>
      <PostalCode>105-0002</PostalCode>
      <TerritoryOfficialName>TOKYO</TerritoryOfficialName>
    </PrimaryAddress>
  </Location>
  <OrganizationDetail>
    <FamilyTreeMemberRole>
      <FamilyTreeMemberRoleText>Subsidiary</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <FamilyTreeMemberRole>
      <FamilyTreeMemberRoleText>Headquarters</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <StandaloneOrganizationIndicator>>false</StandaloneOrganizationIndicator>
  </OrganizationDetail>
  <Linkage>
    <LinkageSummary>
      <ChildrenSummary>
        <ChildrenQuantity>1</ChildrenQuantity>
        <DirectChildrenIndicator>>false</DirectChildrenIndicator>
      </ChildrenSummary>
      <ChildrenSummary>
        <ChildrenTypeText>Affiliate</ChildrenTypeText>
        <ChildrenQuantity>29</ChildrenQuantity>
        <DirectChildrenIndicator>>false</DirectChildrenIndicator>
      </ChildrenSummary>
      <ChildrenSummary>
        <ChildrenTypeText>Branch</ChildrenTypeText>
        <ChildrenQuantity>1</ChildrenQuantity>
        <DirectChildrenIndicator>>true</DirectChildrenIndicator>
      </ChildrenSummary>
      <SiblingCount>29</SiblingCount>
    </LinkageSummary>
    <GlobalUltimateOrganization>
      <DUNSNumber>825320344</DUNSNumber>
    </GlobalUltimateOrganization>
    <DomesticUltimateOrganization>
      <DUNSNumber>697557825</DUNSNumber>
    </DomesticUltimateOrganization>
    <ParentOrganization>
      <DUNSNumber>825320344</DUNSNumber>
    </ParentOrganization>
    <FamilyTreeMemberOrganization>
      <AssociationTypeText>HeadquartersBranch</AssociationTypeText>
      <OrganizationName>
        <OrganizationPrimaryName>

```

```

        <OrganizationName>INFORMATICA JAPAN K.K.</OrganizationName>
    </OrganizationPrimaryName>
</OrganizationName>
<SubjectHeader>
    <DUNSNumber>692640710</DUNSNumber>
    <SubjectHandling>
        <SubjectHandlingText DNBCodeValue="11028">De-listed</
SubjectHandlingText>
    </SubjectHandling>
</SubjectHeader>
<Location>
    <PrimaryAddress>
        <StreetAddressLine>
            <LineText>2-2-2, UMEDA, KITA-KU</LineText>
        </StreetAddressLine>
        <PrimaryTownName>OSAKA</PrimaryTownName>
        <CountryISOAlpha2Code>JP</CountryISOAlpha2Code>
        <TerritoryAbbreviatedName>OSK</TerritoryAbbreviatedName>
        <PostalCode>530-0001</PostalCode>
        <TerritoryOfficialName>OSAKA</TerritoryOfficialName>
    </PrimaryAddress>
</Location>
<OrganizationDetail>
    <FamilyTreeMemberRole>
        <FamilyTreeMemberRoleText>Branch</FamilyTreeMemberRoleText>
    </FamilyTreeMemberRole>
    <StandaloneOrganizationIndicator>>false</StandaloneOrganizationIndicator>
</OrganizationDetail>
<Linkage>
    <GlobalUltimateOrganization>
        <DUNSNumber>825320344</DUNSNumber>
    </GlobalUltimateOrganization>
    <DomesticUltimateOrganization>
        <DUNSNumber>697557825</DUNSNumber>
    </DomesticUltimateOrganization>
    <HeadquartersOrganization>
        <DUNSNumber>697557825</DUNSNumber>
    </HeadquartersOrganization>
    <FamilyTreeHierarchyLevel>2</FamilyTreeHierarchyLevel>
</Linkage>
</FamilyTreeMemberOrganization>
<FamilyTreeHierarchyLevel>2</FamilyTreeHierarchyLevel>
</Linkage>
</FamilyTreeMemberOrganization>

```

Ejemplo de respuesta de API

En la siguiente respuesta de ejemplo se muestran el encabezado y el cuerpo de la respuesta después de importar correctamente una entidad de negocio de organización:

```

BES-interactionId: 72202100242034
BES-processId: 156048
{
  "Organization": {
    "key": {
      "rowid": "101921",
      "sourceKey": "697557825"
    },
    "rowidObject": "101921"
  }
}

```

Actualización de DaaS

La API de REST de actualización de DaaS acepta los datos en formato XML antes y después de un cambio. La API aplica los cambios al registro.

La API usa el método POST.

URL de solicitud

La dirección URL de actualización de DaaS tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/daas/update/  
FamilyTreeMemberOrganizationToOrgView
```

Use la siguiente solicitud HTTP POST para la dirección URL de actualización de DaaS:

```
POST http://<host>:<port>/<context>/<database ID>/daas/update/  
FamilyTreeMemberOrganizationToOrgView
```

Parámetros de consulta

El nombre del sistema de origen es un parámetro de obligatorio.

En la siguiente tabla se indican los parámetros que puede usar en la solicitud:

Parámetro	Descripción
systemName	Obligatorio. Nombre del sistema de origen que realiza el cambio en los datos.
interactionId	Opcional. ID de interacción que se asigna a todos los cambios. Normalmente, el concentrador genera el ID al crear un cambio pendiente como resultado de una llamada.
effectivePeriod	Opcional. Contiene la fecha de inicio y la fecha de finalización. Especifica el periodo de tiempo en el que el registro es efectivo. Especifique estos parámetros para un registro habilitado para la línea temporal.
validateOnly	Opcional. Si se establece en TRUE, solo se aplican las reglas de validación al registro modificado y los cambios no son persistentes.
recordState	Opcional. Estado del concentrador para los registros creados o actualizados. Los estados de registro admitidos son ACTIVE y PENDING.
processId	Opcional. ID del proceso de flujo de trabajo que contiene la tarea. Si el flujo de trabajo se inicia como un resultado de la llamada de servicio, el parámetro contiene el identificador del proceso de flujo de trabajo iniciado.

TEMAS RELACIONADOS

- [“Formatos de fecha y hora UTC” en la página 30](#)

Cuerpo de la solicitud

El cuerpo de la solicitud debe contener un elemento XML o JSON del tipo

DaaSChangeFamilyTreeMemberOrganizationToOrgView en el espacio de nombres urn:co-
ors.informatica.mdm.

Encabezado de respuesta

Cuando la respuesta sea satisfactoria, la API devolverá el ID de interacción (interactionId) y el ID de proceso (processId) en el encabezado de la respuesta y los detalles del registro en el cuerpo de la respuesta.

Si el proceso genera un ID de interacción y lo utiliza para crear el registro, la API devolverá el ID de interacción. Si el proceso inicia un flujo de trabajo en lugar de guardar directamente el registro en la base de datos, la API devolverá el ID de proceso, que es el ID del proceso de flujo de trabajo.

En el ejemplo siguiente se muestra un encabezado de respuesta con un ID de interacción y un ID de proceso:

```
BES-interactionId: 72200000242000
BES-processId: 15948
```

Ejemplo de solicitud de API

La API acepta dos respuestas en formato XML. Una respuesta es anterior a un cambio y la otra respuesta es posterior al cambio. En la siguiente solicitud se añade un número de teléfono nuevo a la organización. Los datos XML *before* no incluyen el número de teléfono y los datos XML *after* sí incluyen el número de teléfono.

La siguiente solicitud contiene el número de teléfono recién añadido:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/daas/update/linkage2org?
systemName=Admin
<urn:DaaSChangelinkage2org xmlns:urn="urn:cs-ors.informatica.mdm" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xsi:type="urn:DaaSChangelinkage2org">
  <urn:before xmlns="http://services.dnb.com/LinkageServiceV2.0">
    <SubjectHeader>
      <DUNSNumber>697557825</DUNSNumber>
    </SubjectHeader>
  </urn:before>

  <urn:after xmlns="http://services.dnb.com/LinkageServiceV2.0">
    <SubjectHeader>
      <DUNSNumber>697557825</DUNSNumber>
    </SubjectHeader>
    <Telecommunication>
      <TelephoneNumber>
        <TelecommunicationNumber>09736250550</TelecommunicationNumber>
        <InternationalDialingCode>1</InternationalDialingCode>
        <UnreachableIndicator>true</UnreachableIndicator>
      </TelephoneNumber>
    </Telecommunication>
  </urn:after>
</urn:DaaSChangelinkage2org>
```

Ejemplo de respuesta de API

En el siguiente ejemplo se muestra el valor de rowId del número de teléfono recién creado de la organización:

```
{
  "Organization": {
    "key": {
      "rowid": "101921",
      "sourceKey": "697557825"
    },
    "rowidObject": "101921",
    "TelephoneNumbers": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "1722",
            "sourceKey": "09736250550"
          },
          "rowidObject": "1722"
        }
      ]
    }
  }
}
```

CAPÍTULO 4

Llamadas de servicios de entidad de negocio de Protocolo simple de acceso a objetos

Este capítulo incluye los siguientes temas:

- [Llamadas de Protocolo simple de acceso a objetos para servicios de entidad de negocio, 133](#)
- [Método de autenticación, 134](#)
- [Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros, 134](#)
- [Archivo de lenguaje de descripción de servicios web, 135](#)
- [URL de SOAP, 136](#)
- [Respuestas y solicitudes SOAP, 137](#)
- [Visualización de parámetros de entrada y salida, 138](#)
- [Referencia de API de SOAP, 139](#)
- [Respuesta y solicitud SOAP de ejemplo, 140](#)

Llamadas de Protocolo simple de acceso a objetos para servicios de entidad de negocio

Las llamadas de extremo de Protocolo simple de acceso a objetos (SOAP) hacen que todos los servicios de entidad de negocio estén disponibles como servicios web. Puede realizar llamadas SOAP para crear, eliminar, actualizar y buscar registros en una entidad de negocio. Puede realizar operaciones como fusionar, anular la fusión y hacer coincidir registros. También puede hacer llamadas SOAP para crear, actualizar, buscar y realizar tareas.

Los extremos de SOAP para servicios de entidad de negocio utilizan el UsernameToken de Seguridad de servicios web (WS-Security) para autenticar a los usuarios.

Nota: Antes de que utilice las API de SOAP para llamar a los servicios de entidad de negocio, valide el Almacén de referencias operativas.

Método de autenticación

Todas las llamadas SOAP a los servicios de entidad de negocio requieren que el usuario se autentique. Proporcione el nombre de usuario y la contraseña en el encabezado de mensaje SOAP de la solicitud de servicio web.

El elemento de encabezado SOAP Security contiene los datos relacionados con la seguridad. El elemento Security contiene el elemento UsernameToken que incluye los siguientes elementos secundarios:

nombre de usuario

El nombre de usuario asociado al token.

contraseña

La contraseña para el nombre de usuario asociado al token.

Envíe el nombre de usuario y la contraseña del elemento UsernameToken.

En el ejemplo siguiente se muestra el elemento de encabezado Security del mensaje SOAP:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <UsernameToken>
        <Username>admin</Username>
        <Password>admin</Password>
      </UsernameToken>
    </Security>
  </soapenv:Header>
<soapenv:Body>
  .....
</soapenv:Body>
</soapenv:Envelope>
```

Cookies de autenticación para el inicio de sesión desde aplicaciones de terceros

Utilice las cookies de autenticación para autenticar los usuarios de MDM Hub y llamar a los servicios de entidades de negocio de aplicaciones de terceros. Puede obtener una cookie basada en las credenciales de un usuario autenticado. Guarde la cookie y utilícela para llamar a la API de SOAP. No es necesario que integre el nombre de usuario y la contraseña como parte del código.

Realice la siguiente solicitud de POST e inicie sesión en la vista Entidad 360 con su nombre de usuario y contraseña:

```
POST http://<host>:<port>/e360/com.informatica.tools.mdm.web.auth/login
{
  user: 'admin',
  password: 'user password'
}
```

Cuando la operación de inicio de sesión se realiza correctamente, el servidor devuelve la cookie de autenticación en el campo de encabezado `set-cookie`. El siguiente código de ejemplo muestra `set-cookie` en el encabezado de respuesta:

```
Set-Cookie: auth_hash_cookie="admin===QTc1RkNGQkNCMzc1RjIyOQ==";
Version=1; Path=/
```

Almacene el código hash y utilícelo en el encabezado de la solicitud de sus llamadas de API. No es necesario que proporcione un nombre de usuario y una contraseña para las llamadas de API.

El siguiente ejemplo muestra cómo se utiliza la cookie de autenticación en el encabezado de solicitud de la API:

```
GET http://<IP of host>/cmx/cs/localhost-orcl-DS_UI1  
Cookie: auth_hash_cookie="admin===QTc1RkNGQkNCMzC1RjIyOQ=="
```

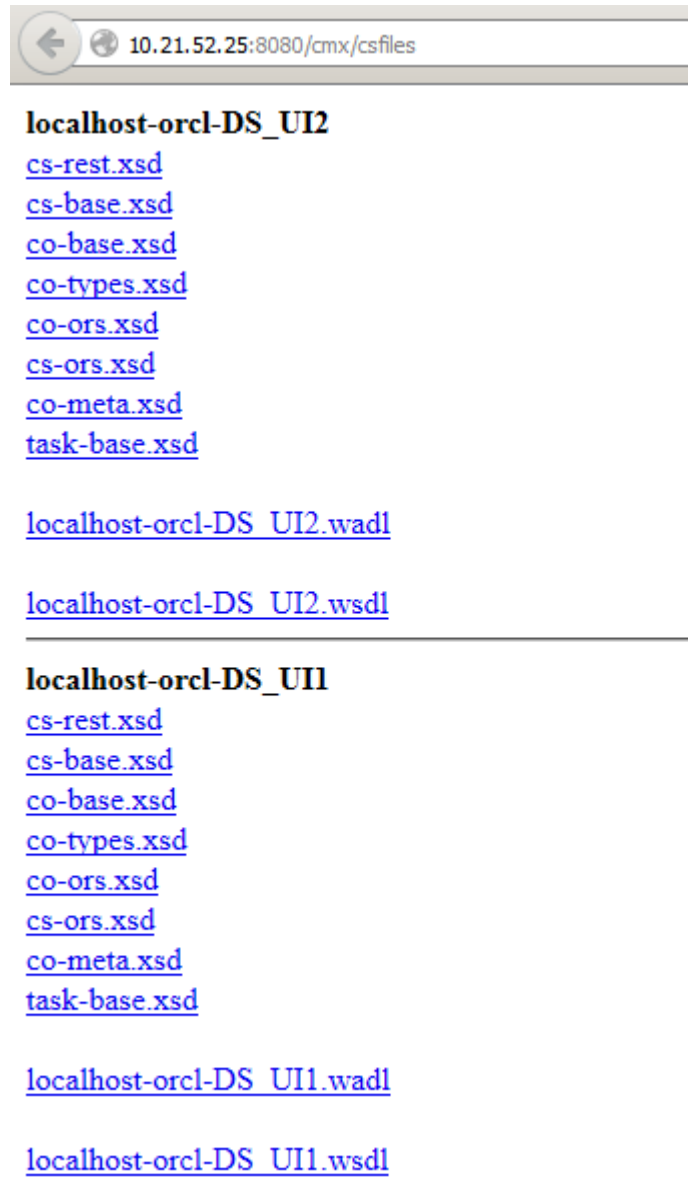
Archivo de lenguaje de descripción de servicios web

Los archivos de lenguaje de descripción de servicios web (WSDL) contienen las descripciones XML de los servicios web, los formatos de las respuestas y las solicitudes SOAP y todos los parámetros. MDM Hub genera un archivo WSDL para cada Almacén de referencias operativas.

Los archivos WSDL para cada Almacén de referencias operativas se encuentran en la siguiente ubicación:

```
http://<host>:<puerto>/cmx/csfiles
```

La siguiente imagen muestra la ubicación en la que puede descargar el archivo WSDL para los almacenes de referencias operativas:



Haga clic en el enlace para descargar el archivo WSDL para el Almacén de referencias operativas DS_UI1 o DS_UI2.

URL de SOAP

Una URL de SOAP es la dirección que utiliza para conectarse al servidor de SOAP.

Una URL de SOAP tiene la siguiente sintaxis:

```
http://<host>:<puerto>/<contexto>/<ID de base de datos>
```


La URL tiene los siguientes campos:

host

El host que ejecuta la base de datos.

puerto

Número de puerto que utiliza el servicio de escucha de la base de datos.

contexto

El contexto siempre es `cmx/services/BEServices`.

ID de base de datos

El ID de ORS como está registrado en la herramienta de bases de datos de la Consola del concentrador.

El siguiente ejemplo muestra una URL de SOAP:

```
http://localhost:8080/cmx/services/BEServices/localhost-orcl-DS_UI1
```

Respuestas y solicitudes SOAP

Utilice el formato de mensaje XML SOAP para enviar solicitudes a través de un cliente SOAP al servicio de entidad de negocio y para recibir respuestas del servicio de entidad de negocio al cliente. El formato de respuesta y solicitud SOAP es el mismo.

El mensaje SOAP contiene los siguientes elementos:

Sobre

Define el inicio y el final del mensaje.

Encabezado

Opcional. Contiene atributos adicionales, como los detalles de autenticación para procesar el mensaje. Si el elemento Header está presente, debe ser el primer elemento secundario del elemento Envelope.

Cuerpo

Contiene los datos XML que procesa el cliente o el servicio web.

Un mensaje SOAP tiene el siguiente formato:

```
<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
</env:Header>

  <env:Body>
</env:Body>

</env:Envelope>
```

Una solicitud SOAP tiene el siguiente formato:

```
POST /<host>:<port>/<context>/<database ID> HTTP/1.0
Content-Type: text/xml; charset=utf-8

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

  <env:Header>
</env:Header>

  <env:Body>
```

```

</env:Body>
</env:Envelope>

```

Una respuesta SOAP tiene el siguiente formato:

```

HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" >

<env:Header>
</env:Header>

<env:Body>
</env:Body>

</env:Envelope>

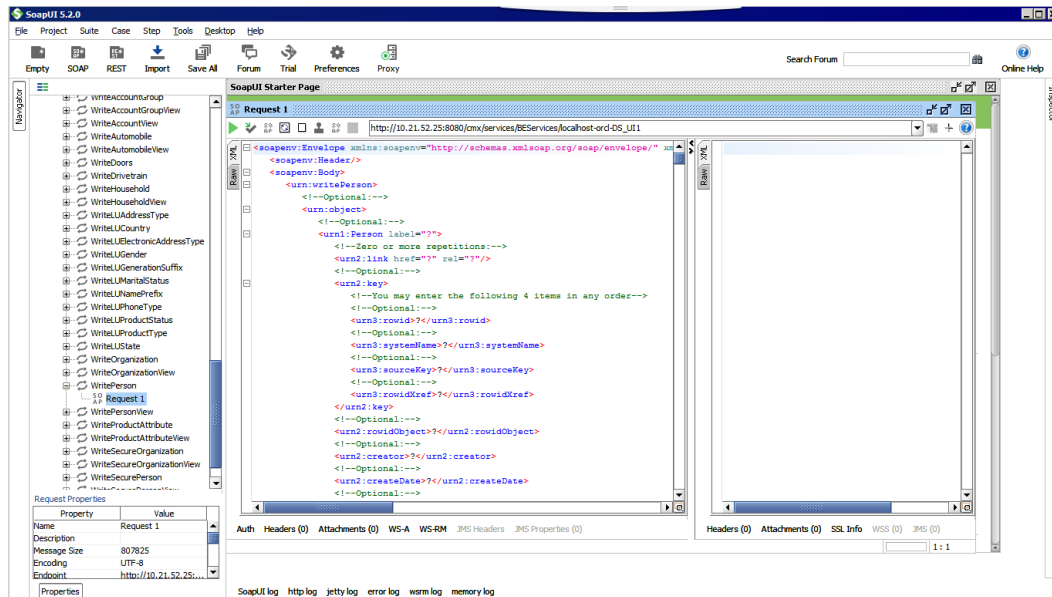
```

Visualización de parámetros de entrada y salida

Puede utilizar una herramienta de pruebas funcional, como SoapUI, para ver los parámetros de entrada y salida de la API de SOAP.

Cree un proyecto de SOAP e importe el archivo WSDL en el proyecto. Las operaciones que puede realizar mediante los servicios de entidad de negocio aparecen como nodos en SoapUI. Cada operación tiene un formato de mensaje de respuesta y de solicitud. SoapUI crea un ejemplo de solicitud para cada operación al importar el archivo WSDL.

Abra el proyecto y haga doble clic en una solicitud para abrir el editor de solicitudes. La siguiente imagen muestra los parámetros de entrada en SoapUI para la API de SOAP WritePerson:



Referencia de API de SOAP

La referencia de la API de SOAP para servicios de entidad de negocio enumera las API de SOAP y proporciona una descripción de cada una de ellas. Consulte también el archivo WSDL para ver las descripciones de los servicios de entidad de negocio.

Utilice las siguientes API de SOAP para realizar operaciones en las entidades de negocio:

Obtener metadatos

Devuelve la estructura de datos de una entidad de negocio.

Enumerar registros

Devuelve la lista de valores de búsqueda o valores de clave externa.

Leer registros

Devuelve los detalles de un registro raíz de la entidad de negocio.

Crear registros

Crea un registro en la entidad de negocio especificada.

Actualizar registros

Actualiza el registro raíz especificado y sus registros secundarios.

Eliminar registros

Elimina un registro raíz en una entidad de negocio.

Buscar registros

Busca un valor de cadena en una entidad de negocio raíz que admite búsquedas y devuelve los registros raíz que coinciden con los criterios de búsqueda.

Vista previa de promoción

Devuelve una vista previa de un registro resultante si promueve los cambios pendientes en función del ID de interacción de la solicitud de cambios.

Promover

Promueve todos los cambios pendientes que se realicen en un registro en función del ID de interacción de la solicitud de cambio.

Eliminar promoción

Elimina todos los cambios pendientes que realice en un registro en función del ID de interacción de la solicitud de cambio.

Vista previa de fusión

Devuelve una vista previa de un registro raíz consolidado si fusiona dos o más registros raíz.

Fusionar registros

Combina dos o más registros raíz para crear un único registro consolidado.

Anular fusión de registros

Anula la fusión de un registro raíz en registros raíz individuales que existían antes de que se fusionaran los registros.

Obtener registros relacionados

Devuelve una lista de los registros relacionados en función de las relaciones que configure en el administrador de jerarquía.

Leer registros coincidentes

Devuelve los registros que coinciden con un registro raíz especificado.

Actualizar registros coincidentes

Crea o actualiza un registro en la tabla de coincidencia.

Eliminar registros coincidentes

Elimina los registros coincidentes de la tabla de coincidencias.

Obtener metadatos de BPM

Devuelve los tipos de tarea y dos indicadores que especifican si el adaptador del flujo de trabajo puede realizar el servicio para obtener el linaje de tarea y los servicios de administración.

Lista de tareas

Devuelve una lista de tareas del flujo de trabajo.

Leer tarea

Devuelve los detalles de una tarea.

Crear tarea

Crea una tarea e inicia un flujo de trabajo.

Actualizar tarea

Actualiza una tarea individual.

Ejecutar acción de tarea

Realiza una acción de tarea y restablece la tarea en el flujo de trabajo para que se siga procesando.

Lista de usuarios asignables

Devuelve una lista de usuarios a los que se les pueden asignar las tareas que pertenecen a un tipo de tarea.

Tarea completada

Cierra un flujo de trabajo de tareas después de completar todas las tareas del flujo de trabajo.

Respuesta y solicitud SOAP de ejemplo

La siguiente solicitud SOAP de ejemplo recupera una lista de los usuarios asignables:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:cs-ors.informatica.mdm">
  <soapenv:Header>
    <Security xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
secext-1.0.xsd">
      <UsernameToken>
        <Username>admin</Username>
        <Password>admin</Password>
      </UsernameToken>
    </Security>
  </soapenv:Header>
  <soapenv:Body>
    <urn:listAssignableUsers>
      <!--Optional:-->
      <urn:parameters>
        <!--Optional:-->
        <urn:taskType>Update</urn:taskType>
    </urn:listAssignableUsers>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <!--Optional:-->
        <urn:businessEntity>Person</urn:businessEntity>
    </urn:parameters>
</urn:listAssignableUsers>
</soapenv:Body>
</soapenv:Envelope>

```

La siguiente respuesta SOAP de ejemplo enumera los usuarios asignables:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <soapenv:Body>
        <ns6:listAssignableUsersReturn xmlns:ns1="urn:cs-base.informatica.mdm"
xmlns:ns2="urn:co-base.informatica.mdm" xmlns:ns3="urn:co-ors.informatica.mdm"
xmlns:ns4="urn:co-meta.informatica.mdm" xmlns:ns5="urn:task-base.informatica.mdm"
xmlns:ns6="urn:cs-ors.informatica.mdm">
            <ns6:object>
                <ns1:users>
                    <user>
                        <userName>admin</userName>
                    </user>
                </users>
                <ns1:roles/>
            </ns6:object>
        </ns6:listAssignableUsersReturn>
    </soapenv:Body>
</soapenv:Envelope>

```

CAPÍTULO 5

Servicios para registros de referencias cruzadas y cálculos de BVT

Este capítulo incluye los siguientes temas:

- [Resumen de los servicios para registros de referencias cruzadas y cálculos de BVT, 142](#)
- [Obtención de datos de referencias cruzadas e investigación de cálculos de BVT, 142](#)
- [Filtrado y paginación de respuestas, 147](#)
- [Establecer la mejor versión de confianza, 147](#)

Resumen de los servicios para registros de referencias cruzadas y cálculos de BVT

Puede usar los servicios para registros de referencias cruzadas y cálculos de la mejor versión de confianza (BVT) para obtener información sobre cómo los datos de origen conforman el registro principal.

Puede usar estos servicios para realizar las siguientes tareas:

- Recopilar información sobre los datos de origen
- Determinar cómo se ha establecido la mejor versión de confianza
- Reemplazar los cálculos de BVT para garantizar que los registros principales contienen la mejor versión de confianza

Obtención de datos de referencias cruzadas e investigación de cálculos de BVT

Los registros principales de MDM Hub mantienen la mejor versión de confianza (BVT). MDM Hub consolida los datos más fiables de varios sistemas de origen en cada registro principal para conseguir la mejor versión de confianza. MDM Hub almacena los datos de origen en registros de referencias cruzadas. Puede usar

servicios de entidad de negocio para leer los datos de los registros de referencias cruzadas y determinar el método de cálculo de BVT.

Obtener registros de referencias cruzadas

Puede usar un servicio de entidad de negocio para obtener los registros de referencias cruzadas para un registro principal concreto.

La dirección URL de la API de REST para la obtención de registros de referencias cruzadas tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=XREF
```

En la siguiente solicitud de ejemplo se recuperan los registros de referencias cruzadas para un registro de entidad de negocio de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=XREF
```

Respuesta generada al obtener registros de referencias cruzadas

En el siguiente ejemplo se muestran los registros de referencias cruzadas que devuelve el registro de persona con el ID de fila 123:

```
GET /Person/123?contentMetadata=XREF

{
  "firstName": "Joe",
  "lastName": "Smith",
  "XREF": {
    "item": [
      {
        "rowidXref": 111,
        "firstName": "Joe",
        "lastName": "Smith",
      },
      {
        "rowidXref": 222,
        "firstName": "John",
        "lastName": "Smith"
      }
    ]
  }
}
```

Determinar colaboradores del registro principal

Puede usar servicios de entidad de negocio para ver qué campo de registro de referencias cruzadas colabora en el registro principal. El registro que colabora en cada campo se identifica por el ID de fila del registro de referencias cruzadas.

La dirección URL de la API de REST para determinar los colaboradores del registro principal tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=BVT
```

En la siguiente solicitud de ejemplo se recupera la información de BVT para un registro de persona con un ID de fila 123:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=BVT
```

Respuesta generada al determinar los colaboradores del registro principal

En el siguiente ejemplo se muestra qué registro de referencias cruzadas ha colaborado en cada campo del registro principal:

```
{
  "firstName": "Joe",
  "lastName": "Smith",
  "BVT": {
    "firstName": {
      "rowidXref": 111
    },
    "lastName": {
      "rowidXref": 222
    }
  }
}
```

Obtener las puntuaciones de confianza de campos de registro de referencias cruzadas colaboradores

Puede usar servicios de entidad de negocio para obtener las puntuaciones de confianza de los campos de registro de referencias cruzadas que colaboran en el registro principal.

La dirección URL de la API de REST para determinar los colaboradores y obtener las puntuaciones de confianza tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=TRUST
```

En la siguiente solicitud de ejemplo se indican las puntuaciones de confianza para un registro de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/ors/Person/123?contentMetadata=TRUST
```

Respuesta generada al obtener las puntuaciones de confianza de campos de registro de referencias cruzadas colaboradores

En el siguiente ejemplo de respuesta se indican las puntuaciones de confianza para cada campo de un registro de entidad de negocio de persona:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "TRUST": {
    "firstName": {
      "score": 75.0,
      "valid": true,
      "trustSetting" :{
        // custom settings
      }
    },
  }
}
```


Obtención de las puntuaciones de confianza de todos los campos de registro de referencias cruzadas

Use la API de REST con el parámetro contentMetadata establecido en XREF_TRUST para obtener las puntuaciones de confianza y los porcentajes de degradación de todos los campos de registro de referencias cruzadas.

Dirección URL de la solicitud para la API de REST de lectura para determinar los colaboradores y obtener las puntuaciones de confianza:

```
http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
contentMetadata=XREF_TRUST
```

En la siguiente solicitud de ejemplo se recuperan los datos de referencias cruzadas para un registro de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?contentMetadata=XREF_TRUST
```

Respuesta generada al obtener las puntuaciones de confianza de todos los campos de registro de referencias cruzadas

En el siguiente ejemplo se muestran las puntuaciones de confianza y los porcentajes de degradación de todos los campos de registro de referencias cruzadas para una entidad de negocio de persona:

```
{
  "firstName": "Sergey",
  "lastName": "Ivanov",
  "XREF": {
    "item": [{
      "rowidXref": 111,
      "firstName": "Sergey",
      "lastName": "Petrov",
      "TRUST": {
        "firstName": {
          "score": 75.0,
          "valid": true
        },
        "lastName": {
          "score": 60.0,
          "valid": false,
          "downgradePerCent": 20.0
        }
      }
    }, {
      "rowidXref": 222,
      "firstName": "Sergey",
      "lastName": "Ivanov",
      "TRUST": {
        "firstName": {
          "score": 10.0,
          "valid": true
        },
        "lastName": {
          "score": 80.0,
          "valid": true
        }
      }
    }
  ]
}
```

Obtener información sobre los sistemas de origen

Puede obtener información sobre los sistemas de origen de los que proceden los datos de referencias cruzadas y determinar el número de registros de referencias cruzadas en los que colaboran los sistemas de origen para todo el registro, para cada nodo o para cada registro.

Los siguientes parámetros se pueden especificar en la solicitud:

describe

Se establece en `true` para devolver la descripción del sistema de origen. Puede ser `true` o `false`. El valor predeterminado es `false`.

aggregate

Define el nivel en el que se devuelve la información de los sistemas de origen. Puede ser `ENTITY`, `NODE` y `RECORD`. El valor predeterminado es `ENTITY`.

recordStates

Especifica el estado de registro para el que se devuelven registros. Puede ser `ACTIVE`, `PENDING` o `DELETED`. El valor predeterminado es `ACTIVE`.

compact

Si se establece en `no`, especifica que los datos de nivel de entidad se devuelven al especificar el parámetro de agregado con `ENTITY` y otros niveles de agregado. Puede ser `yes` o `no`. Solo para solicitudes de API de REST. El valor predeterminado es `yes`.

Ejemplo de obtención de información sobre los sistemas de origen

En la siguiente solicitud de ejemplo se obtiene información sobre los sistemas de origen en el nivel de entidad y el nivel de nodo para la entidad de negocio de persona con el ID de fila 123:

```
GET http://localhost:8080/cmx/cs/ors/Person/123?
action=getSourceSystems&aggregate=ENTITY,NODE&compact=no
```

En la siguiente solicitud de ejemplo se obtienen información sobre los sistemas de origen en el nivel de registro y descripciones de los sistemas de origen para la entidad de negocio de persona con el ID de fila 456:

```
GET http://localhost:8080/cmx/cs/ors/Person/123/Address/456?
action=getSourceSystems&aggregate=ENTITY,NODE&compact=no
```

Respuesta generada al obtener información sobre los sistemas de origen

En el siguiente ejemplo se muestra información de nivel de entidad y de nivel de nodo para una entidad de negocio de persona:

```
{
  "name": "Admin",
  "xrefCount": 120
},
Person: {
  "rowidObject": "456",
  "sourceSystem":
  {
    "name": "Admin",
    "xrefCount": 30
  }
}
```

Filtrado y paginación de respuestas

Puede seleccionar los campos que se van a devolver en la respuesta, filtrar los resultados por varios criterios y paginar los resultados.

Ejemplos de solicitudes de filtrado

En la siguiente tabla se incluyen solicitudes de ejemplo para la entidad de negocio de persona con diversos filtros aplicados y una descripción de los resultados que se devuelven en la respuesta:

Solicitud	Descripción de los resultados devueltos
/Person/123	Todos los campos definidos por el usuario
/Person/123?readSystemFields=true	Todos los campos definidos por el usuario y todos los campos del sistema
/Person/123?fields=firstName	Un solo campo definido por el usuario
/Person/123?fields=updatedAt	Un solo campo del sistema
Person/123?fields=firstName,updatedAt	Un solo campo definido por el usuario y un solo campo del sistema
/Person/123?fields=firstName&readsystemFields=true	Un solo campo definido por el usuario y todos los campos del sistema

Establecer la mejor versión de confianza

Una vez que un gestor de datos haya investigado los datos de origen de los registros de referencias cruzadas, puede realizar ajustes en el modo de consolidación de los datos de origen para garantizar que el registro maestro representa la mejor versión de confianza.

Puede usar servicios de entidad de negocio para realizar las siguientes acciones con el fin de establecer la mejor versión de confianza:

- Actualizar la configuración de confianza
- Eliminar los datos de origen no coincidentes
- Seleccionar el campo colaborador correcto
- Escribir el valor correcto en el registro principal

Seleccionar el campo colaborador correcto

Si el campo con la puntuación de confianza más alta no contiene la mejor versión de confianza, un gestor de datos puede seleccionar el campo que contiene los datos correctos para aportarlos al registro principal.

La dirección URL y el cuerpo de la solicitud para seleccionar el campo colaborador correcto según el nombre del sistema y la clave de origen tienen el siguiente formato:

```
POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system name>
```

```

{
  BVT: {
    <field name>: {
      systemName: "<source system name>",
      sourceKey: "<source key>"
    }
  }
}

```

La dirección URL y el cuerpo de la solicitud para seleccionar el campo colaborador correcto según el ID de registro de referencias cruzadas tienen el siguiente formato:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system name>
{
  BVT: {
    <field name>: {
      rowidXref: "<row ID>"
    }
  }
}

```

Ejemplo de selección del campo colaborador correcto

La siguiente dirección URL y el cuerpo de la solicitud permiten seleccionar el campo de nombre del registro de referencias cruzadas del sistema de origen de ventas con una clave de origen 0001 para colaborar en el registro principal:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Admin
{
  BVT: {
    firstName: {
      systemName: "Sales",
      sourceKey: "0001"
    }
  }
}

```

La siguiente dirección URL y el cuerpo de la solicitud permiten seleccionar el campo de nombre del registro de referencias cruzadas con el ID de fila 789 para colaborar en el registro principal:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Admin
{
  BVT: {
    firstName: {
      rowidXref: "789"
    }
  }
}

```

Escribir el valor correcto en el registro principal

Si usa una llamada de servicio de entidad de negocio para escribir un valor correcto en el registro principal, también puede establecer la configuración de confianza para el valor. Si no especifica la configuración de confianza, MDM Hub usa la configuración del sistema del administrador.

La dirección URL y el cuerpo de la solicitud para escribir el valor correcto con la configuración de confianza del administrador tienen el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system providing the correct value>{
  "<field name>": "<correct value>",
  "$original": {
    "<field name>": "<current value>",
  },
}

```

```

"TRUST": {
  "<field name>": {
    "trustSetting" : {
      custom: false
    }
  }
}

```

La dirección URL y el cuerpo de la solicitud para escribir el valor correcto con la configuración de confianza definida tienen el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
systemName=<source system providing the correct value>{
  "<field name>": "<correct value>",
  "$original": {
    "<field name>": "<current value>",
  },
  "TRUST": {
    "firstName": {
      "trustSetting" : {
        custom: true, // if custom=true, all other trustSetting fields
                      //are mandatory. If they are not set,
                      //the service will return an error.
        minimumTrust: <minimum trust percent>,
        maximumTrust: <maximum trust percent>,
        timeUnit: "<units for measuring trust decay>",
        maximumTimeUnits: <number of units>,
        graphType: "<name of graph type>"
      }
    }
  }
}

```

Parámetros de confianza

Puede definir los siguientes parámetros de confianza:

minimumTrust

Nivel de confianza de un valor de datos cuando es antiguo (una vez transcurrido el período de caída). Este valor debe ser menor o igual que el valor de confianza máximo.

Nota: Si los valores máximo y mínimo de confianza son iguales, la curva de caída será una línea recta y el período y el tipo de caída no tendrán efecto.

maximumTrust

Nivel de confianza de un valor de datos justo después de cambiarlo. Por ejemplo, si el sistema de origen X cambia un campo de número de teléfono de 555-1234 a 555-4321, el nuevo valor tendrá el valor máximo de confianza del sistema X para el campo de número de teléfono. Al configurar un nivel máximo de confianza relativamente alto, se asegura de que los cambios que se realicen en los sistemas de origen se aplicarán al objeto base.

timeUnit

Especifica las unidades que se utilizan para calcular el período de caída: día, semana, mes, trimestre o año.

maximumTimeUnits

Especifica el número (de días, semanas, meses, trimestres o años) que se usa para calcular el período de caída.

graphType

La caída sigue un patrón según el cual el nivel de confianza disminuye durante el período de caída. Los tipos de gráfico pueden ser uno de los siguientes patrones de caída:

Parámetro de tipo de gráfico	Descripción
LINEAR	Es la caída más simple. La caída sigue una línea recta desde el valor máximo de confianza hasta el valor mínimo de confianza.
RISL	La mayor parte de la disminución se produce al inicio del período de caída. La caída sigue una curva cóncava. Si un sistema de origen incluye este tipo de gráfico, el nuevo valor del sistema será de confianza probablemente, pero es posible que dicho valor se reemplace.
SIRL	La mayor parte del aumento se produce al final del período de caída. La caída sigue una curva convexa. Si un sistema de origen incluye este tipo de gráfico, es relativamente poco probable que otro sistema reemplace el valor en el registro principal hasta que el valor esté cerca del final de su período de caída.

Ejemplo de escritura del valor correcto en el registro principal

Ejemplo 1

Desea cambiar el nombre del registro principal de Sam Brown a John Smith. El cambio se atribuye al sistema de origen de ventas. La configuración de confianza se establece en la configuración de confianza del administrador.

En el siguiente código se muestran la dirección URL y el comando para el ejemplo 1.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=Sales
{
  "firstName": "John",
  "lastName": "Smith"
  "$original": {
    "firstName": "Sam",
    "lastName": "Brown"
  },
  "TRUST": {
    "firstName": {
      "trustSetting" : {
        custom: false
      }
    }
    "lastName": {
      "trustSetting" : {
        custom: false
      }
    }
  }
}
```

Ejemplo 2

Desea cambiar el nombre del registro principal de Sam Brown a John Smith. El cambio se atribuye al sistema de origen SFA. La configuración de confianza se establece en un valor de confianza mínimo de 60 y un valor de confianza máximo de 90, y la confianza disminuye de forma lineal durante un período de caída de tres meses.

En el siguiente código se muestran la dirección URL y el comando para el ejemplo 2.

```
POST http://localhost:8080/cmx/cs/localhost-orcl-ORS/Person/123?systemName=SFA
{
  "firstName": "John",
```

```

    "lastName": "Smith"
    "$original": {
      "firstName": "Sam",
      "lastName": "Brown"
    },
    "TRUST": {
      "firstName": {
        "trustSetting" : {
          custom: true,
          minimumTrust: 60,
          maximumTrust: 90,
          timeUnit: "Month",
          maximumTimeUnits: 3,
          graphType: "LINEAR"
        }
      }
      "lastName": {
        "trustSetting" : {
          custom: true,
          minimumTrust: 60,
          maximumTrust: 90,
          timeUnit: "Month",
          maximumTimeUnits: 3,
          graphType: "LINEAR"
        }
      }
    }
  }
}

```

Eliminar los datos de origen no coincidentes

Si un registro de referencias cruzadas está asociado de forma incorrecta a un registro principal determinado, un gestor de datos puede anular la fusión del registro de referencias cruzadas. Se crea, entonces, un nuevo registro principal a partir del registro de referencias cruzadas no fusionado.

Solo se puede anular la fusión de un único registro de referencias cruzadas en una llamada de anulación de fusión. Si es necesario anular la fusión de varios registros de referencias cruzadas, realice una llamada de anulación de fusión para cada registro de referencias cruzadas.

Si se configura un activador para un evento no fusionado, se crea una tarea de anulación de fusión. En caso contrario, se anula la fusión del registro de referencias cruzadas.

La dirección URL y el comando para anular la fusión de un registro basado en el nombre del sistema y la clave de origen tienen el siguiente formato:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
action=unmerge&systemName=<source system name>
{
  name: "<object name>",
  key: {rowid: "<rowid value>", sourcekey: "<source key>", systemName: "<source system
name>" }
}

```

La dirección URL y el comando para anular la fusión de un registro basado en el ID de registro de referencias cruzadas tienen el siguiente formato:

```

POST http://<host>:<port>/<context>/<database ID>/<business entity>/<row ID>?
action=unmerge&systemName=<source system name>
{
  name: "<object name>",
  key: {rowid: "<rowid value>", rowidXref: "<row ID of xref>"}
}

```

Ejemplo de eliminación de datos de origen no coincidentes

Ejemplo de API de REST

En el siguiente código se muestran la dirección URL y el comando para anular la fusión del registro de referencias cruzadas en el nivel secundario de un registro de dirección:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person/181921?
action=unmerge&systemName=Admin
{
  "name": "Person.Address",
  "key": {
    "rowid": "41721 ",
    "rowidXref": 41722
  }
}
```

Donde:

- El registro de referencias cruzadas para el que se va a anular la fusión tiene el ID de fila 41722.
- El ID de fila del registro principal desde el que se va a anular la fusión del registro de referencias cruzadas es 41721.
- El ID de fila del registro raíz es 181921.

Ejemplo de SOAP/EJB

En el siguiente código se muestran la dirección URL y el comando para anular la fusión del registro de referencias cruzadas en el nivel secundario de un registro de dirección:

```
<ns9:UnMerge xmlns:ns2="urn:co-base.informatica.mdm" xmlns:ns7="urn:co-
meta.informatica.mdm" xmlns:ns3="http://services.dnb.com/LinkageServiceV2.0"
xmlns:ns8="urn:task-base.informatica.mdm" xmlns:ns6="urn:co-ors.informatica.mdm"
xmlns:ns1="urn:cs-base.informatica.mdm" xmlns:ns9="urn:cs-ors.informatica.mdm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="ns9:UnMerge">
  <ns9:parameters>
    <ns9:businessEntityKey name="Person">
      <ns1:key>
        <ns1:rowid>181921</ns1:rowid>
      </ns1:key>
    </ns9:businessEntityKey>
    <ns9:unmergeKey name="Person.TelephoneNumbers">
      <ns1:key>
        <ns1:rowid>41721 </ns1:rowid>
        <ns1:rowidXref>41722</ns1:rowidXref>
      </ns1:key>
    </ns9:unmergeKey>
    <ns9:treeUnmerge>true</ns9:treeUnmerge>
  </ns9:parameters>
</ns9:UnMerge>
```

Donde:

- El registro de referencias cruzadas para el que se va a anular la fusión tiene el ID de fila 41722.
- El ID de fila del registro principal desde el que se va a anular la fusión del registro de referencias cruzadas es 41721.
- El ID de fila del registro raíz es 181921.

Respuesta de anulación de fusión

La respuesta de anulación de fusión contiene el ID de fila del objeto base que se crea a partir del registro de referencias cruzadas sin fusionar.

Ejemplo de respuesta 1

En el siguiente ejemplo se muestra la respuesta si usa un registro de referencias cruzadas de un nodo raíz de persona.

```
{
  Person: {
    rowidObject: "7777"
  }
}
```

Ejemplo de respuesta 2

En el siguiente ejemplo se muestra la respuesta si usa un registro de referencias cruzadas de un nodo secundario de persona.

```
{
  Person: {
    Address: {
      item: [
        rowidObject: "55555"
      ]
    }
  }
}
```

CAPÍTULO 6

Compatibilidad del servicio de vinculación empresarial

Este capítulo incluye los siguientes temas:

- [Resumen, 154](#)
- [Servicios de entidad de negocio para la importación y actualización de DaaS, 154](#)
- [Configuración de la compatibilidad de la vinculación, 155](#)
- [Aplicación personalizada para la división de datos de vinculación, 155](#)

Resumen

El servicio de vinculación empresarial de Duns & Bradstreet (D&B) devuelve el elemento principal de una organización solicitada y todas sus entidades relacionadas. Puede usar el servicio de vinculación de D&B para obtener información de todas las sucursales y divisiones de una organización. Puede crear y actualizar registros con los datos del servicio de vinculación.

Puede importar los datos de vinculación empresarial en MDM Hub. Debe desarrollar una aplicación personalizada que pueda usar el servicio de vinculación del componente personalizado de proveedor de DaaS en una vista de entidad.

Necesita un servicio de entidad de negocio para importar los datos del servicio de D&B y crear un registro con los datos. En caso de producirse un cambio en los datos almacenados externamente, debe poder realizar los cambios correspondientes en su registro. D&B ofrece un servicio de supervisión para informarle de los cambios en los datos. Necesita un servicio que pueda aceptar datos antes y después de un cambio, y aplicar los cambios al registro correspondiente.

Servicios de entidad de negocio para la importación y actualización de DaaS

El servicio de entidad de negocio de importación de DaaS acepta datos del servicio de vinculación en formato XML y los convierte en un registro. El servicio de entidad de negocio de actualización de DaaS acepta datos del servicio externo en forma de dos archivos XML. Estos dos archivos XML se corresponden

con los datos antes y después de un cambio. El servicio de actualización aplica los cambios al registro correspondiente.

TEMAS RELACIONADOS

- [“Importación de DaaS” en la página 127](#)
- [“Actualización de DaaS” en la página 130](#)

Configuración de la compatibilidad de la vinculación

Con el fin de poder usar el servicio de vinculación de D&B para crear y actualizar registros, debe añadir la configuración en la herramienta de aprovisionamiento y crear una aplicación personalizada para dividir la respuesta del servicio de vinculación.

Realice las siguientes tareas para configurar la compatibilidad del servicio de vinculación de D&B:

1. Use la herramienta de aprovisionamiento para cargar WSDL para el servicio de vinculación.
2. Use la herramienta de aprovisionamiento para configurar un documento XML para la transformación de entidad de negocio y su exposición como un servicio. Cuando expone la transformación como un servicio, el proceso crea los servicios de importación y actualización de DaaS.
3. Cree una aplicación personalizada que pueda solicitar datos del servicio de vinculación y dividir la respuesta en detalles de registro y detalles de vinculación.
4. Desarrolle una interfaz de usuario que envíe la solicitud a la aplicación personalizada.

Nota: Para obtener más información sobre cómo cargar WSDL y configurar XML para la transformación de entidad de negocio, consulte el capítulo Integración de Data as a Service de la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

Aplicación personalizada para la división de datos de vinculación

Para poder usar los servicios de vinculación de D&B, debe diseñar una aplicación personalizada que pueda dividir la información de vinculación en detalles de registro y detalles de vinculación.

La aplicación personalizada debe realizar las siguientes funciones:

1. Aceptar la solicitud para un servicio de vinculación en la vista de entidad.
2. Enviar la solicitud a D&B y recibir la respuesta.
3. Convertir la respuesta en XML.
4. Dividir la respuesta en detalles de registro y detalles de vinculación.
5. Enviar la información XML a servicios de entidad de negocio para guardarla como un registro en la base de datos.
6. Supervisar los cambios en los datos y llamar a la función de aviso de cambio en lista del servicio externo.

CAPÍTULO 7

Llamadas externas para limpiar, analizar y transformar datos

Este capítulo incluye los siguientes temas:

- [Resumen, 156](#)
- [Eventos compatibles, 157](#)
- [Cómo configurar llamadas externas, 157](#)
- [Ejemplo: Validación y lógica personalizadas para servicios de entidad de negocio, 158](#)

Resumen

Los proveedores externos proporcionan servicios web para limpiar, analizar y transformar datos de registro. Use los servicios web externos para la validación personalizada (por ejemplo, para comprobar si el campo de dirección está en blanco al añadir un registro). Use los servicios web externos para que la lógica personalizada transforme los datos de registro. Por ejemplo, si fusiona dos registros, puede fusionar las direcciones, pero no los números de teléfono.

Un servicio web externo expone una o varias operaciones a las que pueden llamar los servicios de entidad de negocio. Cada operación tiene un tipo de solicitud y respuesta. Los servicios de entidad de negocio envían los datos de registro con los parámetros de servicio requeridos para los servicios externos. Puede configurar llamadas a los servicios web externos para determinados pasos de la lógica de ejecución. En función de la lógica que implemente, las solicitudes salen de Data Director para actualizar los datos de registro. Es posible que los servicios externos modifiquen los datos si es necesario.

En la herramienta de aprovisionamiento, configure la entidad de negocio y los eventos para los que desea llamar al servicio externo. En la herramienta de aprovisionamiento, cargue el archivo WSDL para el servicio externo y registre el servicio SOAP y la operación. Enlace el servicio con entidades de negocio y eventos específicos.

Use el archivo WSDL del kit de recursos para comprender el servicio, las operaciones, los métodos y los tipos de datos que intercambian los métodos de servicio. El archivo `custom-logic-service.wsdl` para los servicios web externos se encuentra en la siguiente ubicación del kit recursos: `C:\<infamd installation directory>\hub\resourcekit\samples\BESEExternalCall\source\resources\webapp\WEB-INF\wsdl\`

El kit de recursos incluye código de ejemplo para implementar la lógica y la validación personalizadas. Si instala el kit de recursos, el archivo `bes-external-call.ear` para la lógica y la validación personalizadas de ejemplo se implementa en el servidor de aplicaciones.

Eventos compatibles

Un servicio de entidad de negocio consta de una serie de pasos de servicio. Puede aplicar la lógica y la validación personalizadas a cualquiera de los pasos.

Puede hacer llamadas externas para los siguientes eventos:

- WriteCO.BeforeEverything
- WriteCO.BeforeValidate
- WriteCO.AfterValidate
- WriteCO.AfterEverything
- WriteView.BeforeEverything
- WriteView.BeforeValidate
- WriteView.AfterValidate
- WriteView.AfterEverything
- MergeCO.BeforeEverything
- MergeCO.AfterEverything
- PreviewMergeCO.BeforeEverything
- PreviewMergeCO.AfterEverything
- ReadCO.BeforeEverything
- ReadCO.AfterEverything
- ReadView.BeforeEverything
- ReadView.AfterEverythingEvents

Cómo configurar llamadas externas

Un servicio de entidad de negocio cuenta con pasos de servicio. Una solicitud entrante pasa a través de cada paso de servicio. Puede configurar llamadas a servicios externos para determinados pasos de la lógica de ejecución de servicio de entidad de negocio.

Realice los siguientes pasos para configurar las llamadas externas:

1. Cree e implemente el archivo `bes-external-call.ear`.
2. En la herramienta de aprovisionamiento, realice las siguientes tareas:
 - a. Cargue el archivo WSDL para el servicio externo.
 - b. Registre el servicio web como un servicio SOAP.
 - c. Configure una llamada externa.

Para obtener más información sobre la carga del archivo WSDL, el registro del servicio SOAP y la configuración de las llamadas externas, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

Para obtener más información sobre la creación e implementación del archivo EAR, consulte la *Guía del kit de recursos de Multidomain MDM*.

Ejemplo: Validación y lógica personalizadas para servicios de entidad de negocio

Puede probar la validación y la lógica personalizadas cuando añada y fusione los registros de persona. La validación personalizada comprueba si el registro de persona tiene una dirección. La lógica personalizada no permite fusionar dos números de teléfono. Use las API de REST para crear y fusionar registros de persona.

1. Para comprobar la validación al crear un registro de persona, realice los siguientes pasos:
 - a. Use la API de creación para crear un registro de persona sin ninguna dirección. Se genera un error de validación.
 - b. Use la API de creación para crear un registro de persona con una dirección. La operación se realiza correctamente.
2. Para comprobar la lógica personalizada al fusionar registros, realice los siguientes pasos:
 - a. Use la API de creación para crear dos registros de persona con direcciones y números de teléfono.
 - b. Use la API de vista previa de fusión para fusionar los dos registros de persona. Añada `overrides` a la solicitud de vista previa de fusión para fusionar las direcciones y los teléfonos. En la respuesta se muestran una sola dirección y dos números de teléfono. La lógica personalizada impide la fusión de los números de teléfono.

Requisitos previos

Para comprobar la lógica y la validación personalizadas, debe cargar el archivo WSDL en la herramienta de aprovisionamiento. Debe registrar el servicio SOAP y la operación. Enlace el servicio con las entidades de negocio y los eventos para los que desea usar la lógica y la validación personalizadas. Puede probar la lógica y la validación para las entidades de negocio y los eventos especificados.

Paso 1. Probar la validación personalizada

Use la API de creación para crear el siguiente registro de persona sin ninguna dirección:

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-mdm_Sample/Person?systemName=Admin
{
    firstName: "John"
}
```

Se genera un error de validación.

Use la API de creación para crear el siguiente registro de persona con una dirección:

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-mdm_Sample/Person?systemName=Admin
{
    firstName: "John",
    Addresses: {
        item: [
            {
                cityName: "Toronto"
            }
        ]
    }
}
```

La solicitud crea un registro de persona.

Paso 2. Probar la lógica personalizada

Realice los siguientes pasos para probar la lógica personalizada:

1. Use la API de creación para crear dos registros de persona con direcciones y teléfonos:

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-mdm_sample/Person?systemName=Admin
{
  firstName: "John",
  Addresses: {
    item: [
      {
        cityName: "Toronto"
      }
    ]
  },
  TelephoneNumbers: {
    item: [
      {
        phoneNum: "111-11-11"
      }
    ]
  }
}
```

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-mdm_sample/Person?systemName=Admin
{
  firstName: "John",
  Addresses: {
    item: [
      {
        cityName: "Ottawa"
      }
    ]
  },
  TelephoneNumbers: {
    item: [
      {
        phoneNum: "222-22-22"
      }
    ]
  }
}
```

La respuesta contiene los siguientes valores de rowid:

- Persona: 161923, 161924
- Direcciones: 2123, 2124
- Números de teléfono: 101723, 101724

2. Ejecute la API de vista previa de fusión para fusionar ambos registros de persona:

```
POST http://localhost:8080/cmxc/cs/localhost-orcl-mdm_sample/Person/161923?
action=previewMerge&depth=2
{
  keys: [
    {
      rowid: "161924"
    }
  ]
}
```

La respuesta es un registro de persona con dos direcciones y dos números de teléfono.

```
{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
  }
}
```

```

"consolidationInd": 4,
"lastRowidSystem": "SYS0",
"hubStateInd": 1,
"label": "Person: , Bill",
"partyType": "Person",
"displayName": "Bill",
"firstName": "Bill",
"TelephoneNumbers": {
  "link": [],
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 2,
  "item": [
    {
      "rowidObject": "101723",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:35.904-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "PhoneNumbers",
      "phoneNum": "111-1111",
      "phoneCountryCd": "1"
    },
    {
      "rowidObject": "101724",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:54.768-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "PhoneNumbers",
      "phoneNum": "222-2222",
      "phoneCountryCd": "1"
    }
  ]
},
"Addresses": {
  "link": [],
  "firstRecord": 1,
  "recordCount": 2,
  "pageSize": 2,
  "item": [
    {
      "rowidObject": "2123",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:37.956-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "Addresses",
      "Address": {
        "rowidObject": "2121",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:36.922-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:37.923-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Address",
        "cityName": "Toronto"
      }
    }
  ]
},

```



```

    {
      "rowidObject": "2124",
      "creator": "admin",
      "createDate": "2016-10-20T09:50:54.790-04:00",
      "updatedBy": "admin",
      "lastUpdateDate": "2016-10-20T09:50:54.790-04:00",
      "consolidationInd": 4,
      "lastRowidSystem": "SYS0",
      "hubStateInd": 1,
      "label": "Addresses",
      "Address": {
        "rowidObject": "2122",
        "creator": "admin",
        "createDate": "2016-10-20T09:50:54.777-04:00",
        "updatedBy": "admin",
        "lastUpdateDate": "2016-10-20T09:50:54.777-04:00",
        "consolidationInd": 4,
        "lastRowidSystem": "SYS0",
        "hubStateInd": 1,
        "label": "Address",
        "cityName": "Ottawa"
      }
    }
  ]
}

```

3. Ejecute la API de vista previa de fusión para fusionar ambos registros de persona, lo que reemplaza la fusión de las direcciones y los números de teléfono:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person/161923?
action=previewMerge&depth=3

```

```

{
  keys: [
    { rowid: "161923" }
  ],
  overrides: {
    Person: {
      Addresses: {
        item: [
          {
            rowidObject: "2123",
            MERGE: {
              item: [{key:{rowid: "2124"}}],
              $original: {
                item:[null]
              }
            }
          }
        ]
      },
      TelephoneNumbers: {
        item: [
          {
            rowidObject: "101723",
            MERGE: {
              item: [{key:{rowid: "101724"}}],
              $original: {
                item:[null]
              }
            }
          }
        ]
      }
    }
  }
}

```

En la respuesta se muestran una sola dirección y dos números de teléfono.

```
{
  "Person": {
    "rowidObject": "161923",
    "creator": "admin",
    "createDate": "2016-10-20T09:50:35.878-04:00",
    "updatedBy": "admin",
    "lastUpdateDate": "2016-10-20T09:50:35.879-04:00",
    "consolidationInd": 4,
    "lastRowidSystem": "SYS0",
    "hubStateInd": 1,
    "label": "Person: , Bill",
    "partyType": "Person",
    "displayName": "Bill",
    "firstName": "Bill",
    "TelephoneNumbers": {
      "link": [],
      "firstRecord": 1,
      "recordCount": 2,
      "pageSize": 2,
      "item": [
        {
          "rowidObject": "101723",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:35.904-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:35.905-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "111-1111",
          "phoneCountryCd": "1"
        },
        {
          "rowidObject": "101724",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:54.768-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:54.769-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "PhoneNumbers",
          "phoneNum": "222-2222",
          "phoneCountryCd": "1"
        }
      ]
    },
    "Addresses": {
      "link": [],
      "firstRecord": 1,
      "recordCount": 1,
      "pageSize": 1,
      "item": [
        {
          "rowidObject": "2123",
          "creator": "admin",
          "createDate": "2016-10-20T09:50:37.956-04:00",
          "updatedBy": "admin",
          "lastUpdateDate": "2016-10-20T09:50:37.956-04:00",
          "consolidationInd": 4,
          "lastRowidSystem": "SYS0",
          "hubStateInd": 1,
          "label": "Addresses"
        }
      ]
    },
    "PersonDetails": {
```

```
    "link": [],  
    "recordCount": 0,  
    "pageSize": 0,  
    "item": []  
  }  
}
```

APÉNDICE A

Uso de las API de REST para añadir registros

Este apéndice incluye los siguientes temas:

- [Resumen sobre el uso de las API de REST para añadir registros, 164](#)
- [Estructura de una entidad de negocio de persona, 165](#)
- [Paso 1. Obtener información sobre el esquema, 165](#)
- [Paso 2. Crear un registro, 171](#)
- [Paso 3. Leer el registro, 173](#)

Resumen sobre el uso de las API de REST para añadir registros

Después de crear un modelo de entidad de negocio y configurar la estructura de entidad de negocio, puede usar las API de REST para añadir registros.

Las secciones siguientes utilizan el ejemplo de la entidad de negocio de persona para ilustrar cómo se pueden añadir registros mediante las API de REST. La entidad de negocio de persona contiene datos de los empleados de su empresa.

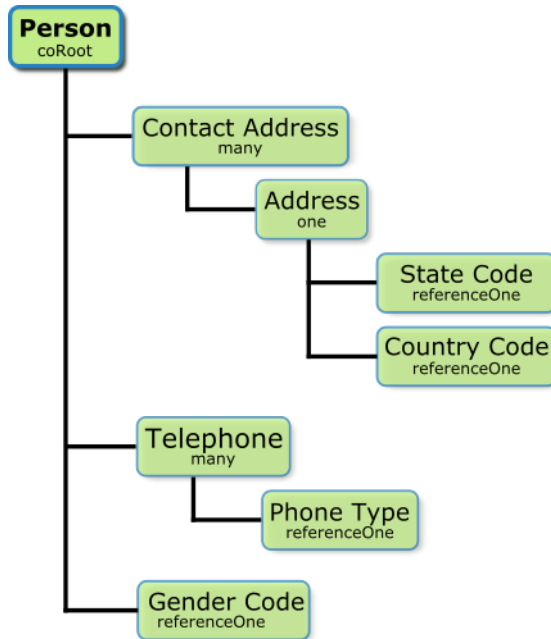
Utilice las siguientes API para añadir detalles de sus empleados:

1. Obtenga información sobre el esquema. Utilice la API de REST Obtener metadatos para conseguir información sobre la estructura de datos de una entidad de negocio, incluida la estructura, la lista de campos, los tipos de campos y los detalles de los campos de búsqueda. O puede acceder a los archivos de definición de esquema XML (XSD) que describen qué elementos y atributos se pueden utilizar. Los archivos XSD residen en `http://<host>:<puerto>/cmx/csfiles`.
2. Cree un registro. Utilice la API de REST de creación de registros para crear el registro.
3. Lea los datos del registro que haya añadido. Utilice la API de REST de lectura de registros para recuperar los datos del registro.

Estructura de una entidad de negocio de persona

Agregaremos un registro de persona mediante las API de REST. El nodo raíz de Persona es el nodo que ocupa la posición más alta en la estructura de entidades de negocio de persona. Debajo del nodo raíz Persona existen nodos para los detalles de los empleados, como el sexo, la dirección y el teléfono.

La siguiente imagen muestra la estructura de la entidad de negocio de Persona:



Persona es el nodo raíz de la entidad de negocio de Persona. El tipo de nodo, que aparece debajo del nombre de nodo, indica la relación entre el nodo superior y el nodo secundario. Existe una relación de uno a uno entre la Dirección de contacto y la Dirección, que indica que cada dirección de contacto solo puede tener asociada una dirección. Existe una relación de uno a varios entre la Persona y el Teléfono, que indica que un registro de persona puede tener asociados varios números de teléfono. Existe una relación de uno a uno entre la Persona y el Sexo, que indica que un registro de persona solo puede tener un valor de sexo. Los valores de sexo residen en una tabla de búsqueda. Del mismo modo, los valores del código de provincia/estado y de país residen en varias tablas de búsqueda.

Paso 1. Obtener información sobre el esquema

Utilice la API de REST para la obtención de metadatos para conseguir información sobre un esquema. La API para la obtención de metadatos devuelve la estructura de datos de una entidad de negocio. Los metadatos enumeran los campos de entidad de negocio, los tipos de campos y los detalles de los campos de búsqueda.

La URL para la obtención de metadatos tiene el siguiente formato:

```
http://<host>:<port>/<context>/<database ID>/<business entity>?action=meta
```

La siguiente solicitud de ejemplo recupera la información de metadatos para la entidad de negocio de persona:

```
GET http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?action=meta
```

Obtener la respuesta de los metadatos

El siguiente ejemplo muestra algunos fragmentos de la estructura de datos de la entidad de negocio de persona:

```
{
  "object": {
    "field": [
      {
        "allowedValues": [
          "Person"
        ],
        "name": "partyType",
        "label": "Party Type",
        "dataType": "String",
        "length": 255,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "imageUrl",
        "label": "Image URL",
        "dataType": "ImageURL",
        "length": 255,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "statusCd",
        "label": "Status Cd",
        "dataType": "String",
        "length": 2,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "displayName",
        "label": "Display Name",
        "dataType": "String",
        "length": 200,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "birthdate",
        "label": "Birthdate",
        "dataType": "Date",
        "length": 0,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
      },
      {
        "name": "firstName",
        "label": "First Name",
        "dataType": "String",
        "length": 50,

```

```

        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "lastName",
        "label": "Last Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "middleName",
        "label": "Middle Name",
        "dataType": "String",
        "length": 50,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "dirtyIndicator",
        "label": "Dirty Indicator",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "hubStateInd",
        "label": "Hub State Ind",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "cmDirtyInd",
        "label": "Content metadata dirty Ind",
        "dataType": "Integer",
        "length": 38,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    },
    {
        "name": "lastRowidSystem",
        "label": "Last Rowid System",
        "dataType": "String",
        "length": 14,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": true,
        "required": false,
        "system": true
    }

```

```

    },
    -----
    {
        "name": "genderCd",
        "label": "Gender Cd",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "lookup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_uil/LUGender?action=list&idlabel=genderCode%3AgenderDisp",
                    "rel": "lookup"
                },
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_uil/LUGender?action=list",
                    "rel": "list"
                }
            ],
            "object": "LUGender",
            "key": "genderCode",
            "value": "genderDisp"
        }
    }
},
    -----

```

```

    "child": [
        {
            "field": [
                {
                    "name": "cityName",
                    "label": "City Name",
                    "dataType": "String",
                    "length": 100,
                    "totalDigits": 0,
                    "fractionDigits": 0,
                    "readOnly": false,
                    "required": false,
                    "system": false
                },
                {
                    "name": "addressLine2",
                    "label": "Address Line2",
                    "dataType": "String",
                    "length": 100,
                    "totalDigits": 0,
                    "fractionDigits": 0,
                    "readOnly": false,
                    "required": false,
                    "system": false
                },
                {
                    "name": "addressLine1",
                    "label": "Address Line1",
                    "dataType": "String",
                    "length": 100,
                    "totalDigits": 0,
                    "fractionDigits": 0,
                    "readOnly": false,
                    "required": false,
                    "system": false
                },
                {
                    "name": "isValidInd",

```



```

        "label": "Is Valid Ind",
        "dataType": "String",
        "length": 1,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    },
    {
        "name": "postalCd",
        "label": "Postal Cd",
        "dataType": "String",
        "length": 10,
        "totalDigits": 0,
        "fractionDigits": 0,
        "readOnly": false,
        "required": false,
        "system": false
    }
},

```

```

    {
        "name": "countryCode",
        "label": "Country Code",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "dependents": [
            "Person.Address.Address.stateCd"
        ],
        "lookup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry?action=list",
                    "rel": "list"
                },
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry?action=list&idlabel=countryCode%3AcountryNameDisp",
                    "rel": "lookup"
                }
            ],
            "object": "LUCountry",
            "key": "countryCode",
            "value": "countryNameDisp"
        }
    },
    {
        "name": "stateCd",
        "label": "State Cd",
        "dataType": "lookup",
        "readOnly": false,
        "required": false,
        "system": false,
        "parents": [
            "Person.Address.Address.countryCode"
        ],
        "lookup": {
            "link": [
                {
                    "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/LUCountry/{Person.Address.Address.countryCode}/LUState?action=list",
                    "rel": "list"
                }
            ],
            "object": "LUState",
            "key": "stateCd",
            "value": "stateNameDisp"
        }
    }
}

```

```

                "href": "http://localhost:8080/cmx/cs/
localhost-hub101-ds_ui1/LUCountry/{Person.Address.Address.countryCode}/LUState?
action=list&idlabel=stateAbbreviation%3AstateNameDisp",
                "rel": "lookup"
            }
        ],
        "object": "LUCountry.LUState",
        "key": "stateAbbreviation",
        "value": "stateNameDisp"
    }
}
],
"name": "Address",
"label": "Address",
"many": false
}
],
"name": "Address",
"label": "Contact Address",
"many": true
},
{
    "field": [
        {
            "name": "phoneNum",
            "label": "Phone Number",
            "dataType": "String",
            "length": 13,
            "totalDigits": 0,
            "fractionDigits": 0,
            "readOnly": false,
            "required": false,
            "system": false
        },

```

```

        {
            "name": "phoneTypeCd",
            "label": "Phone Type",
            "dataType": "lookup",
            "readOnly": false,
            "required": false,
            "system": false,
            "lookup": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-
hub101-ds_ui1/LUPhoneType?action=list&idlabel=phoneType%3AphoneTypeDisp",
                        "rel": "lookup"
                    },
                    {
                        "href": "http://localhost:8080/cmx/cs/localhost-
hub101-ds_ui1/LUPhoneType?action=list",
                        "rel": "list"
                    }
                ],
                "object": "LUPhoneType",
                "key": "phoneType",
                "value": "phoneTypeDisp"
            }
        },
        "name": "Telephone",
        "label": "Telephone",
        "many": true
    }
],
"name": "Person",
"label": "Person",

```

```

    "many": false
  }
}

```

Paso 2. Crear un registro

Utilice la API de REST de creación de registros para crear un registro. El nombre de la entidad de negocio y el nombre del sistema de origen son parámetros obligatorios. Envíe los datos para el registro en el cuerpo de la solicitud.

La dirección URL de creación de registros tiene el siguiente formato:

```

http://<host>:<port>/<context>/<database ID>/<business entity>?systemName=<name of the
source system>

```

El parámetro `systemName` es un parámetro obligatorio y especifica el nombre del sistema de origen.

La entidad de negocio de persona tiene el nodo raíz `Persona` y los nodos de dirección de segundo nivel, sexo y teléfono.

La siguiente solicitud de ejemplo crea un registro de persona:

```

POST http://localhost:8080/cmxc/cs/localhost-hub101-ds_uil/Person?systemName=Admin
{
  "firstName": "Boris",
  "lastName": "Isaac",
  "genderCd": {
    "genderCode": "M"
  },
  "Address": {
    "item": [
      {
        "Address": {
          "addressLine1": "B-203, 101 Avenue, New York",
          "stateCd": {
            "stateAbbreviation": "NY"
          },
          "countryCode": {
            "countryCode": "US"
          }
        }
      }
    ]
  },
  "Telephone": {
    "item": [
      {
        "phoneNum": "1234567",
        "phoneTypeCd": {
          "phoneType": "HOM"
        }
      },
      {
        "phoneNum": "7654321",
        "phoneTypeCd": {
          "phoneType": "MOB"
        }
      }
    ]
  }
}

```

El cuerpo de la solicitud especifica los detalles siguientes del registro de persona:

- Nombre.

- Apellido.
- Sexo.
- Dirección con el código de provincia/estado y el código de país.
- Números de teléfono junto con el tipo de teléfono como, por ejemplo, fijo o móvil.

Crear respuestas de registro

La siguiente respuesta de ejemplo muestra la respuesta después de crear correctamente un registro de persona:

```
{
  "Person": {
    "key": {
      "rowid": "658248",
      "sourceKey": "66240000025000"
    },
    "rowidObject": "658248",
    "genderCd": {
      "key": {
        "rowid": "2"
      },
      "rowidObject": "2"
    },
    "Address": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "101526",
            "sourceKey": "66240000028000"
          },
          "rowidObject": "101526",
          "Address": {
            "key": {
              "rowid": "121506",
              "sourceKey": "66240000027000"
            },
            "rowidObject": "121506",
            "countryCode": {
              "key": {
                "rowid": "233"
              },
              "rowidObject": "233"
            },
            "stateCd": {
              "key": {
                "rowid": "52"
              },
              "rowidObject": "52"
            }
          }
        }
      ]
    },
    "Telephone": {
      "link": [],
      "item": [
        {
          "key": {
            "rowid": "20967",
            "sourceKey": "66240000029000"
          },
          "rowidObject": "20967",
          "phoneTypeCd": {
            "key": {

```


Nota: Si un flujo de trabajo está iniciado cuando crea un registro, este tendrá el estado pendiente. De forma predeterminada, la solicitud de lectura de registros lee los registros activos. Utilice el parámetro `recordStates` para especificar el estado pendiente del registro.

La siguiente solicitud de ejemplo lee los detalles de un registro consolidado:

```
GET http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248?
depth=3&recordStates=PENDING
```

Leer la respuesta del registro

La siguiente respuesta de ejemplo muestra los detalles del registro que ha añadido:

```
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248?
depth=2",
      "rel": "children"
    }
  ],
  "rowidObject": "658248",
  "label": "Person",
  "partyType": "Person",
  "displayName": "BORIS ISAAC",
  "firstName": "BORIS",
  "lastName": "ISAAC",
  "genderCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/genderCd/2",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/genderCd/2?depth=2",
        "rel": "children"
      }
    ]
  },
  "rowidObject": "2",
  "label": "LU Gender",
  "genderCode": "M",
  "genderDisp": "MALE"
},
"Address": {
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248/Address",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/
658248",
      "rel": "parent"
    }
  ]
},
"firstRecord": 1,
"pageSize": 10,
```

```

"searchToken": "SVR1.PCWJ",
"item": [
  {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/
Person/658248/Address",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/
Person/658248/Address/101526?depth=2",
        "rel": "children"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/
Person/658248/Address/101526",
        "rel": "self"
      }
    ],
    "rowidObject": "101526",
    "label": "Contact Address",
    "Address": {
      "link": [
        {
          "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506?depth=2",
          "rel": "children"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526",
          "rel": "parent"
        },
        {
          "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506",
          "rel": "self"
        }
      ],
      "rowidObject": "121506",
      "label": "Address",
      "addressLine1": "B-203, 101 Avenue, New York",
      "countryCode": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/countryCode",
            "rel": "self"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506",
            "rel": "parent"
          },
          {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/countryCode?depth=2",
            "rel": "children"
          }
        ],
        "countryCode": "US"
      },
      "stateCd": {
        "link": [
          {
            "href": "http://localhost:8080/cmx/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506",
            "rel": "parent"
          }
        ]
      }
    }
  }
]

```

```

                "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/stateCd?depth=2",
                "rel": "children"
            },
            {
                "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Address/101526/Address/121506/stateCd",
                "rel": "self"
            }
        ],
        "stateAbbreviation": "NY"
    }
}
]
},
"Telephone": {
    "link": [
        {
            "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/Person/
658248",
            "rel": "parent"
        },
        {
            "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/Person/
658248/Telephone",
            "rel": "self"
        }
    ],
    "firstRecord": 1,
    "pageSize": 10,
    "searchToken": "SVR1.PCWK",
    "item": [
        {
            "link": [
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone",
                    "rel": "parent"
                },
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone/20967",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8080/cmxc/cs/localhost-hub101-ds_ui1/
Person/658248/Telephone/20967?depth=2",
                    "rel": "children"
                }
            ],
            "rowidObject": "20967",
            "label": "Telephone",
            "phoneNum": "1234567",
            "phoneTypeCd": {
                "link": [
                    {
                        "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967",
                        "rel": "parent"
                    },
                    {
                        "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967/phoneTypeCd/8",
                        "rel": "self"
                    },
                    {
                        "href": "http://localhost:8080/cmxc/cs/localhost-hub101-
ds_ui1/Person/658248/Telephone/20967/phoneTypeCd/8?depth=2",
                        "rel": "children"
                    }
                ]
            }
        }
    ]
}

```



```

    ],
    "rowidObject": "8",
    "label": "LU Phone Type",
    "phoneTypeDisp": "HOME",
    "phoneType": "HOM"
  }
},
{
  "link": [
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968",
      "rel": "self"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968?depth=2",
      "rel": "children"
    },
    {
      "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone",
      "rel": "parent"
    }
  ],
  "rowidObject": "20968",
  "label": "Telephone",
  "phoneNum": "7654321",
  "phoneTypeCd": {
    "link": [
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968/phoneTypeCd/6",
        "rel": "self"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968",
        "rel": "parent"
      },
      {
        "href": "http://localhost:8080/cmx/cs/localhost-hub101-ds_ui1/Person/658248/Telephone/20968/phoneTypeCd/6?depth=2",
        "rel": "children"
      }
    ]
  },
  "rowidObject": "6",
  "label": "LU Phone Type",
  "phoneTypeDisp": "MOBILE",
  "phoneType": "MOB"
}
]
}
}
}

```

APÉNDICE B

Uso de las API de REST para cargar archivos

Este apéndice incluye los siguientes temas:

- [Resumen sobre el uso de las API de REST para cargar archivos, 178](#)
- [API de REST para archivos, 179](#)
- [Componentes de archivo, 179](#)
- [Tipos de almacenamiento, 180](#)
- [Adjuntar archivos a registros, 181](#)
- [Adjuntar archivos a tareas, 182](#)
- [Cargar archivos de paquete de recursos, 185](#)

Resumen sobre el uso de las API de REST para cargar archivos

Puede usar las API de REST para cargar archivos en un tipo de almacenamiento. Tras cargar un archivo, puede adjuntarlo a un registro o a una tarea, o bien puede usarlo para localizar la interfaz de usuario de Data Director.

En función de cómo quiera usar los archivos, podría variar la combinación de las API de REST, los componentes de archivo y el tipo de almacenamiento que utilice. Por ejemplo, para adjuntar archivos a registros o tareas, debe crear los metadatos del archivo y crear el archivo en un almacenamiento temporal. Tras cargar el archivo, puede adjuntarlo a un registro en una base de datos o adjuntarlo a una tarea en el almacenamiento BPM. Para localizar la interfaz de usuario de Data Director, descargue el archivo ZIP, modifique los archivos comprimidos y, a continuación, cargue el paquete ZIP modificado en el almacenamiento BUNDLE.

API de REST para archivos

Es posible usar un conjunto de diversas API de REST con fines generales para cargar y administrar archivos para adjuntos o su localización.

En la siguiente tabla se enumeran las API de REST para los archivos:

API de REST	Descripción	Tipo de almacenamiento compatible
Enumerar metadatos de archivo	Devuelve una lista de los metadatos de archivo en un almacenamiento.	BPM o TEMP
Crear metadatos de archivo	Crea los metadatos de un archivo en un almacenamiento.	DB o TEMP
Obtener metadatos de archivo	Devuelve los metadatos de un archivo.	BPM, BUNDLE, DB o TEMP
Actualizar metadatos de archivo	Actualiza los metadatos de un archivo.	DB o TEMP
Cargar contenido de archivo	Carga el contenido de un archivo en un almacenamiento.	BUNDLE, DB o TEMP
Obtener contenido de archivo	Descarga el contenido de un archivo.	BPM, BUNDLE, DB o TEMP
Suprimir archivo	Elimina un archivo en un almacenamiento, incluidos los componentes asociados al archivo, como su contenido o sus metadatos.	BUNDLE, DB o TEMP

Componentes de archivo

Para adjuntar archivos a registros o tareas, cree metadatos para el archivo y, a continuación, cargue el contenido del archivo. Para localizar la interfaz de usuario de Data Director, descargue el archivo de paquete de recursos y, a continuación, cargue el archivo de paquete de recursos que se ha modificado.

Metadatos de archivo

Información sobre el archivo, como el nombre del archivo, el tipo de archivo y su contenido. En función del tipo de almacenamiento, es posible que se tengan que incluir otros parámetros, como el creador, la hora de creación y la fecha de carga.

Contenido de archivo

El contenido del archivo. Por ejemplo, el texto, la imagen, el documento o el paquete de recursos.

Tipos de almacenamiento

Cargue y almacene archivos en una implementación de almacenamiento compatible. El tipo de almacenamiento que utilice dependerá de si quiere localizar la interfaz de usuario de Data Director o adjuntar archivos a registros o tareas.

En la siguiente lista se describen los tipos de almacenamiento compatibles:

BPM

Almacena archivos adjuntos a tareas junto con los datos de la tarea. Cuando adjunte un archivo a una tarea, el proceso almacenará el archivo en un almacenamiento BPM del almacenamiento TEMP.

Los archivos almacenados en el almacenamiento BPM utilizan el formato de ID de archivo siguiente:
`taskId::filename.`

Nota: Para adjuntar un archivo a tareas activadas o tareas existentes, en la herramienta de aprovisionamiento, permita los adjuntos en activadores de tarea, tipos de tarea y acciones de tarea. Para obtener más información, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

BUNDLE

Almacena archivos de paquete de recursos que localiza la interfaz de usuario de Data Director.

Los archivos almacenados en el almacenamiento BUNDLE utilizan el formato de ID de archivo siguiente:
`besMetadata.`

DB

Almacena adjuntos de archivo para registros en la tabla C_REPOS_ATTACHMENTS. Cuando adjunte un archivo a un registro, el proceso almacenará el archivo en un almacenamiento DB del almacenamiento TEMP.

Los archivos almacenados en el almacenamiento DB utilizan el formato de ID de archivo siguiente:
`DB_<RowID>.`

Nota: Para adjuntar un archivo a un registro, en la herramienta de aprovisionamiento, configure un campo con `FileAttachment` como el tipo de dato. Si desea más información sobre cómo se configura el tipo de dato, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

TEMP

Almacena temporalmente los archivos en la tabla C_REPOS_ATTACHMENTS y marca los archivos como TEMP. Los archivos se eliminan del almacenamiento TEMP después de que se hayan cargado correctamente al almacenamiento BPM o DB o después de que haya pasado el tiempo de caducidad preconfigurado.

Los archivos almacenados en el almacenamiento TEMP utilizan el formato de ID de archivo siguiente:
`TEMP_<ROWID_ATTACHMENT>.`

Si desea más información sobre cómo se configura el tiempo de caducidad, consulte la *Guía de configuración de Multidomain MDM*.

Adjuntar archivos a registros

Antes de adjuntar un archivo a un registro, cree los metadatos del archivo y, a continuación, cargue el archivo en el almacenamiento temporal.

1. Para crear los metadatos de un archivo, utilice la API de REST de creación de metadatos de archivos con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se crean los metadatos del archivo `Document_3.pdf`:

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP
Content-Type: application/json
{
  "fileName": "Document_3.pdf",
  "fileType": "pdf",
  "fileContentType": "application/pdf"
}
```

Nota: Cree siempre los metadatos de archivos en el almacenamiento TEMP.

La API de REST de creación de metadatos de archivos devuelve un ID para el archivo. El ID de archivo se presenta en el siguiente formato: `<Tipo de almacenamiento>_<RowID>`. Donde RowID hace referencia al ID de fila del archivo que se va a cargar en el almacenamiento.

En el ejemplo, la llamada de API devuelve el siguiente ID para el archivo `Document_3.pdf`:

```
TEMP_SVR1.OJU3
```

El ID de archivo se puede usar para cargar el archivo, adjuntarlo, actualizarlo, descargarlo y eliminarlo.

2. Para cargar el archivo, utilice la API de REST de carga de contenido de archivo con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se carga el archivo en el almacenamiento TEMP:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.OJU3/
content
Content-Type: application/octet-stream
<file object (upload using REST client)>
```

Nota: Después de cargar un archivo, el almacenamiento TEMP almacena el archivo durante un período de tiempo preconfigurado de 60 minutos. El archivo se debe adjuntar a un registro antes de que caduque el período preconfigurado.

3. Para crear un registro y adjuntar el archivo a un registro nuevo, utilice la API de REST de creación de registros.

Por ejemplo, con la siguiente solicitud se crea un registro y se adjunta el archivo con el ID de archivo

```
TEMP_SVR1.OJU3:
```

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/Person?systemName=Admin
Content-Type: application/json
{
  "frstNm": "John",
  "lstNm": "Smith",
  "addrLn1": "2100 Breverly Road",
  "addrTyp": {
    "addrTyp": "Billing",
    "addrTypDesc": "Billing"
  },
  "cntryCd": {
    "cntryCd": "AX",
    "cntryDesc": "Aland"
  },
  "attachments": {
    "item": [
      {
        "fileId": "TEMP_SVR1.OJU3"
      }
    ]
  }
}
```

```
}  
}
```

Nota: Cuando adjunte un archivo a un registro, el proceso almacenará el archivo en la base de datos. El ID del archivo cambiará a DB_<RowID>, donde DB indica que el archivo se encuentra almacenado en la base de datos.

4. Para reemplazar un archivo que está adjunto a un registro, utilice la API de REST de carga de contenido de archivo con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se reemplaza el archivo adjunto con el ID de archivo

DB_SVR1.OJU3 de la base de datos:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3/content  
Content-Type: application/octet-stream  
<file object (upload using REST client)>
```

Nota: El tipo de almacenamiento de la URL de la solicitud es DB.

5. Para editar los metadatos de un archivo después de adjuntarlo a un registro, utilice la API de REST de actualización de metadatos de archivo con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se actualizan los metadatos de un archivo asociado al ID de archivo DB_SVR1.OJU3 de la base de datos:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3  
Content-Type: application/json  
{  
  "fileName": "Document_4.pdf",  
  "fileType": "pdf",  
  "fileContentType": "application/pdf"  
}
```

6. Para descargar un archivo que está adjunto a un registro, utilice la API de REST para obtener el contenido de archivo con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se descarga un archivo que está asociado al ID de archivo

DB_SVR1.OJU3 desde la base de datos:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3/content
```

7. Para eliminar un archivo que está adjunto a un registro, utilice la API de REST de eliminación de archivos con DB como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se elimina un archivo que está asociado al ID de archivo

DB_SVR1.OJU3 de la base de datos:

```
DELETE http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/DB/DB_SVR1.OJU3
```

Adjuntar archivos a tareas

Cree los metadatos de un archivo y, a continuación, cargue el contenido del archivo en un almacenamiento temporal. Después de cargar el archivo, adjúntelo a una tarea activada o a una tarea existente.

Nota: Para adjuntar un archivo a tareas activadas o tareas existentes, en la herramienta de aprovisionamiento, permita los adjuntos en activadores de tarea, tipos de tarea y acciones de tarea. Para obtener más información, consulte la *Guía de la herramienta de aprovisionamiento de Multidomain MDM*.

1. Para crear los metadatos de un archivo, utilice la API de REST de creación de metadatos de archivos con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se crean los metadatos del archivo file1.txt:

```
POST http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP  
  
{  
  "fileName": "file1.txt",
```

```

    "fileType": "text",
    "fileContentType": "text/plain"
  }

```

Nota: Cree siempre los metadatos de archivos en el almacenamiento TEMP.

La API de REST de creación de metadatos de archivos devuelve un ID para el archivo. El ID de archivo se presenta en el siguiente formato: <Tipo de almacenamiento>_<RowID>. Donde RowID hace referencia al ID de fila del archivo que se va a cargar en el almacenamiento.

En el ejemplo, la llamada de API devuelve el siguiente ID para file1.txt: TEMP_SVR1.1VDVS

El ID de archivo se puede usar para cargar el archivo, adjuntarlo, actualizarlo y eliminarlo.

2. Para cargar el archivo, utilice la API de REST de carga de contenido de archivo con TEMP como tipo de almacenamiento.

Por ejemplo, con la siguiente solicitud se carga el archivo en el almacenamiento TEMP:

```

PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/TEMP/TEMP_SVR1.1VDVS/
content

Test attachment content: file 1

```

Nota: Después de cargar un archivo, el almacenamiento TEMP almacena el archivo durante un período de tiempo preconfigurado de 60 minutos. El archivo se debe adjuntar a una tarea antes de que caduque el período preconfigurado.

3. Adjunte el archivo a la tarea que se activó mientras administraba los registros.
 - Para adjuntar el archivo a la tarea que se activa al crear un registro, use la API de REST de creación de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se crea un registro y se adjunta el archivo con el ID de archivo TEMP_SVR1.1VDVS:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person?
systemName=Admin&taskAttachments=TEMP_SVR1.1VDVS
Content-Type: application/json

{
  firstName: "John",
  lastName: "Smith",
  Phone: {
    item: [
      {
        phoneNumber: "111-11-11"
      }
    ]
  }
}

```

- Para adjuntar el archivo a la tarea que se activa al actualizar un registro, use la API de REST de actualización de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se actualiza un registro y se adjunta el archivo con el ID de archivo TEMP_SVR1.1VDVS:

```

PUT http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/233?
systemName=Admin&taskAttachments=TEMP_SVR1.1VDVS
{
  rowidObject: "233",
  firstName: "BOB",
  lastName: "LLOYD",
  Phone: {
    item: [
      {
        rowidObject: "164",
        phoneNumber: "777-77-77",
        $original: {
          phoneNumber: "(336) 366-4936"
        }
      }
    ]
  }
}

```

```

    }
  ]
},
$original: {
  firstName: "DUNN"
}
}

```

- Para adjuntar el archivo a la tarea que se activa al fusionar un registro, use la API de REST de fusión de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se fusiona un registro y se adjunta el archivo con el ID de archivo `TEMP_SVR1.1VDVS`:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478245?
action=merge&taskAttachments=TEMP_SVR1.1VDVS
Content-Type: application/<json/xml>
{
  keys: [
    {
      rowid: "2478246"
    }
  ],
  overrides: {
    Person: {
      firstName: "Charlie"
    }
  }
}

```

- Para adjuntar el archivo a la tarea que se activa al anular la fusión de un registro, use la API de REST de anulación de fusión de entidades de negocio con el parámetro `taskattachments`.

Por ejemplo, con la siguiente solicitud se anula la fusión de un registro y se adjunta el archivo con el ID de archivo `TEMP_SVR1.1VDVS`:

```

POST http://localhost:8080/cmx/cs/localhost-orcl-DS_UI1/Person/2478248?
action=unmerge&taskAttachments=TEMP_SVR1.1VDVS
{
  rowid: "4880369"
}

```

4. Adjunte el archivo a una tarea existente.

- Para adjuntar el archivo al actualizar una tarea, use la API de REST de actualización de tareas con `adjuntos` en el cuerpo de la solicitud.

Por ejemplo, con la siguiente solicitud se actualiza una tarea y se adjunta el archivo con el ID de archivo `TEMP_SVR1.1VDVS`:

```

PUT http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/task/urn:b4p2:15934
{
  taskType: {
    name: "UpdateWithApprovalWorkflow"
  },
  taskId: "urn:b4p2:15934",
  owner: "John",
  title: "Smoke test task - updated",
  comments: "Smoke testing - updated",
  "attachments": [
    {
      "id": "TEMP_SVR1.1VDVS"
    }
  ],
  ...
}

```

- Para adjuntar el archivo al ejecutar una acción de tarea, use la API de REST de ejecución de acciones de tarea con `adjuntos` en el cuerpo de la solicitud.

Por ejemplo, con la siguiente solicitud se ejecuta una acción de tarea y se adjunta el archivo con el ID de archivo TEMP_SVR1.1VDVS:

```
POST http://localhost:8080/cmx/cs/localhost-orcl-MDM_SAMPLE/task/urn:b4p2:15934?
taskAction=Cancel
{
  taskType: {
    name:"UpdateWithApprovalWorkflow",
    taskAction: [{name: "Cancel"}]
  },
  taskId: "urn:b4p2:15934",
  owner: "manager",
  title: "Smoke test task 222",
  comments: "Smoke testing",
  "attachments": [
    {
      "id": "TEMP_SVR1.1VDVS"
    }
  ],
  ...
}
```

Después de adjuntar un archivo a una tarea, el proceso mueve el archivo del almacenamiento TEMP y lo almacena junto a los datos de tarea en el almacenamiento de BPM. El ID del archivo cambia a taskId::filename.

Cargar archivos de paquete de recursos

Para localizar la interfaz de usuario de Data Director, descargue el archivo ZIP de paquete de recursos, modifique los archivos del archivo ZIP y, a continuación, cargue el archivo ZIP modificado en el almacenamiento BUNDLE.

1. Para descargar el archivo ZIP de paquete de recursos, utilice la API de REST de obtención de contenido de archivo con BUNDLE como tipo de almacenamiento.

Por ejemplo, la siguiente solicitud descarga el archivo ZIP de paquete de recursos:

```
GET http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/
content
```

2. Modifique el archivo ZIP añadiendo archivos de paquete específicos de idioma.
Por ejemplo, para localizar los nombres de campo, las etiquetas y los nombres de las tablas a ruso, añada el archivo `besMetadata_ru.properties`.

3. Para cargar el archivo ZIP de paquete de recursos modificado, utilice la API de REST de carga de contenido de archivo con BUNDLE como tipo de almacenamiento.

Por ejemplo, la siguiente solicitud carga el archivo ZIP de paquete de recursos:

```
PUT http://localhost:8080/cmx/file/localhost-orcl-MDM_SAMPLE/BUNDLE/besMetadata/
content
Content-Type: application/octet-stream
Body: binary stream - zip file with besMetadata bundle
```

INDICE

A

- actualizar registros
 - cuerpo de la respuesta [54](#)
 - encabezado de respuesta [54](#)
 - parámetros de URL [52](#)
- API de REST
 - Actualización de DaaS [130](#)
 - actualizar metadatos de archivo [87](#)
 - actualizar registros [52](#)
 - actualizar registros coincidentes [113](#)
 - actualizar tarea [76](#)
 - actualizar una relación [105](#)
 - anular fusión de registros [100](#)
 - buscar registros [58](#)
 - búsqueda de DaaS [120](#)
 - crear metadatos de archivo [85](#)
 - crear registros [50](#)
 - crear tarea [73](#)
 - crear una relación [104](#)
 - cuerpo [27](#)
 - cuerpo de la solicitud [28](#)
 - ejecutar acción de tarea [81](#)
 - eliminar pendiente [92](#)
 - eliminar registros [55](#)
 - eliminar registros coincidentes [113](#)
 - eliminar una relación [107](#)
 - encabezado [27](#)
 - encabezado de solicitud [27](#)
 - enumerar metadatos de archivo [84](#)
 - enumerar registros [56](#)
 - fusión pendiente [96](#)
 - fusionar registros [98](#)
 - Importación de DaaS [127](#)
 - lectura de DaaS [125](#)
 - leer registros [43](#)
 - leer relación [101](#)
 - leer tarea [71](#)
 - lista de metadatos [38](#)
 - lista de tareas [66](#)
 - lista de usuarios asignables [83](#)
 - obtener contenido de archivo [89](#)
 - obtener detalles de evento [117](#)
 - obtener eventos de historial de registros [114](#)
 - obtener metadatos [35](#)
 - obtener metadatos de archivo [86](#)
 - obtener metadatos de BPM [65](#)
 - obtener metadatos de DaaS [119](#)
 - obtener registros coincidentes [111](#)
 - obtener registros relacionados [107](#)
 - promover fusión [97](#)
 - promover registro [92](#)
 - proveedor de sugerencias [64](#)
 - suprimir archivo [89](#)
 - tarea completada [79](#)
 - vista previa de fusión [93](#)

- API de REST (*continuado*)
 - vista previa de promoción [90](#)
- API SOAP
 - autenticación [134](#)
 - respuesta [137](#)
 - solicitud [137](#)
 - WSDL [135](#)
- autenticación
 - HTTP básico [24](#)
 - método [24](#)
 - uso de cookies [25](#), [134](#)

C

- confianza
 - WriteBE [10](#)
- configurar
 - llamadas externas [157](#)
- crear registros
 - cuerpo de la respuesta [51](#)
 - encabezado de respuesta [51](#)
 - parámetros de URL [50](#)
 - URL de solicitud [50](#)
- crear tarea
 - URL de solicitud [73](#)
- Cuerpo de REST
 - Formato JSON [29](#)
 - formato XML [28](#)

D

- datos de vinculación
 - aplicación personalizada [155](#)
 - dividir [155](#)

E

- ejecutar acción de tarea
 - cuerpo de la solicitud [81](#)
- ejemplo
 - llamadas externas [158](#)
 - lógica personalizada [158](#)
- eliminar registros
 - parámetro de URL [56](#)
 - URL de solicitud [55](#)
- eventos compatibles
 - list [157](#)

F

- formato de fecha
 - acerca de [30](#)

L

- leer registros
 - parámetros de consulta [45](#)
- leer tarea
 - URL de solicitud [71](#)
- lista de tareas
 - parámetros de clasificación [67](#)
 - parámetros de consulta [66](#)
 - URL de solicitud [66](#)
- llamadas externas
 - resumen [156](#)

M

- métodos REST
 - compatibles [24](#)
 - DELETE [24](#)
 - GET [24](#)
 - PATCH [24](#)
 - POST [24](#)
 - PUT [24](#)

O

- obtener detalles de evento
 - parámetros de consulta [117](#)
- obtener eventos de historial de registros
 - parámetros de consulta [115](#)

P

- parámetros de consulta
 - depth [29](#)
 - firstRecord [29](#)
 - recordsToReturn [29](#)
 - returnTotal [29](#)
 - searchToken [29](#)
- pasos del servicio de entidad de negocio
 - SearchBE [11](#)
 - WriteBE [10](#)
- periodos efectivos
 - WriteBE [10](#)
- probar llamadas externas
 - requisitos previos [158](#)
- prólogo [7](#)

R

- ReadBE
 - servicio de entidad de negocio [10](#)

- registro eliminado
 - restaurar [10](#)
- registros
 - añadir [164](#)
 - uso de las API de REST [164](#)
- registros raíz
 - identificación [12](#)
- restaurar
 - registro eliminado parcialmente [10](#)

S

- SearchBE
 - acerca de [11](#)
- servicio de entidad de negocio
 - ReadBE [10](#)
- servicio de vinculación
 - configuración [155](#)
- servicios de entidad de negocio
 - Actualización de DaaS [155](#)
 - extremo de SOAP [12](#)
 - Extremo EJB [11](#)
 - Extremo REST [11](#)
 - extremos [11](#)
 - extremos de SOAP [133](#)
 - Extremos REST [23](#)
 - Importación de DaaS [155](#)
 - Referencia de la API de REST [35](#)

U

- UTC
 - acerca de [30](#)

V

- vinculación empresarial
 - compatibilidad [154](#)

W

- WriteBE
 - paso del servicio de entidad de negocio [10](#)

Z

- zona horaria
 - acerca de [30](#)